SYSTEM REFERENCE GUIDE

HACC/VM 1.4.1

HOST
AML
COMMUNICATION
CONTROL

Form No.: DOC V00 010

Version vom 8. Juni 1999

SYSTEM REFERENCE GUIDE FOR HACC/VM 1.4.1

IMPRINT

This manual was written with utmost care. Textual or formal errors can still not be excluded!

Protected trademarks are not marked as such in this manual. The fact that these trademarks are not shown does not imply that the trade names are free for use.

Subject to changes without notice.

Publisher: ADIC/GRAU Storage Systems GmbH
Eschenstraße 3
89558 Böhmenkirch

Production: GRAU Software GmbH,
Gottlieb-Daimler-Straße 17/3
74385 Pleidelsheim

LITERATURE

Manual	Order number German	Order number English	Reference
System Reference Guide	DOC V00 009	DOC V00 010	G
Operator Guide	DOC V00 006	DOC V00 007	G
Installation and Customization	DOC V00 008	DOC V00 011	G

CONTENTS

FIGURES

INTRODUCTION

PRINCIPLES OF AML INTEGRATION UNDER VM

The AML robot systems are controlled by the operating system/platform VM/VSE through the host component, HACC/VM. **HACC** (**H**OST AML COMMUNICATION **C**ONTROL) serves as interface between tape management applications of the VM system or the VSE guest system as well as an automation tool for various administrative tasks during cartridge processing.

Figure 1: AML Integration under VM

AML INTEGRATION UNDER VM/CMS

The following figure depicts integration of the CMS into the AML robot system and its interaction with the HACC/VM host software.

Figure 2: AML Integration under VM/CMS

AML INTEGRATION UNDER VM/CMS

The following figure depicts integration of the VSE guest systems with the appropriate tape management into the AML robot systems and its interaction with the HACC/VM host software.

Figure 3: AML integration under VM/VSE

THE HACC/VM HOST SOFTWARE

For the integration of AML under VM, the following components are distinguished to provide software solutions in the host environment:

Figure 4: Component overview

HACC/VM CLIENT/SERVER INSTALLATION

The HACC/VM function is realized by a client/server installation with the following topology:

- In each connected (attached) VM system a HACC/VM **server** has been installed which is responsible for the central management of all cartridge operations performed by the robot of the VM system.
- The following types are distinguished as **clients** of this server machine:

Client Type	Function
SUBSYSTEM	Virtual machines (VSE, CMS) running applications which process cartridges of the robot system. These machines generate Mount/Keep requests for the HACC/VM.
ADAPTER	Communication Interface between HACC/VM and the AMU (AML Management Unit) which controls the AML system
MONITOR	Process observation and console
ROUTER	Interface between cartridge processing applications and HACC/VM (message processing interface)

The number of respective HACC/VM clients is not limited.

HACC/VM operates entirely independent and autonomous as a closed system.

SUBSYSTEM COMMUNICATION INTERFACE (SSCI)

The communication interface between HACC/VM and subsystems like VSE and CMS consists of individual components depending upon the tape management system:

■ HACC/VSE	• Interface between a specific HACC/VM tape management exit and the HACC/VM Server machine • Drive control after task completion
■ Tape management exit	HACC/VM component (OPEN, CLOSE & MESSAGE) used for DYNAM/T
■ VSE additional product	e.g. FAQS/ASO for the integration of EPIC/VSE
■ Router	for linkage to HACC/VM controlled by console message

In principle these components are routines depending on the respective carrier system (supervisor) and the respective tape management system used.

SUBSYSTEMS WITH CARTRIDGE PROCESSING

The respective subsystems are supplemented with adequate SSCI components depending upon the tape management system. This provides the basis for communication with the autonomous component, HACC/VM.

The guest systems running under VM form the operational environment for tape cartridge processing and thus the basis for the use of AML robot systems, e.g.:

- VSE/ESA + Tape management system

  and/or

- CMS under VM/ESA

The requests for mounting/dismounting that occur under the control of these subsystems (called MOUNT and KEEP in HACC terminology) need appropriate handling.

If these requests (Mount/Keep) refer to drives to be operated by the robot system (AML), the subsystem requests have to be transmitted at the "AMU" interface. This function is performed by HACC/VM components.

For this purpose the following HACC/VM components and/or virtual machines have to be activated:

- SERVER machine
- Subsystem machine(s) Client type SUBSYSTEM
- ADAPTER machine(s) Client type ADAPTER
- ROUTER machine Client type MONITOR (optional)

As a rule the existence of an operable tape management system within the subsystems is assumed. On the VSE side this is mandatory, because VSE is communicating with HACC/VM through a robot interface (HACC/VSE or an interface integrated into the tape management system), the architecture of which requires a tape management system.

Other subsystems (non VSE) can possibly be integrated without a tape management system.

A detailed description of the possible subsystem integrations, including tape management systems, can be found in the chapter *Integration of* beginning on page 1.

The use of the tape management systems DYNAM/T and EPIC/VSE in a VSE system under VM also require the installation of the SSCI component HACC/VSE[1] within the VSE system.

Further information on this topic can be found in the chapter *VSE* beginning on page 2.

In contrast to VSE, the support of the CMS environment does not necessarily require a tape management system. In this case the cartridges can be processed either on the basis of message interception (ROUTER) or by direct communication between the respective CMS machine and HACC/VM using HACC/VM commands (see *VM* beginning on page 31 and the *HACC/VM Operator Guide*).

This means that the CMS machine which initiates cartridge processing has to transmit unique messages for Mount/Keep. HACC/VM is able to accept messages in any format[2], which requires the consideration of the following criteria:

- Mounting and dismounting (Mount/Keep)
- Virtual or real unit address
  - Specific unit address
  - Unit pool (e.g. ANY)
- Volser (mandatory only for Mount)
  - Specific volsers
  - Volser pool (e.g. SCRATCH)

[1] For integration of EPIC/VSE the component HACC/VSE is not mandatory. However, the use of HACC/VSE is recommended, to guarantee the release of available drives of the AML robot system.

[2] The format of the request messages of the subsystem can be configured by adapting the appropriate HACC/VM parameter (S04$SUBS) or by special filter programs.

CONFIGURATIONS

A HACC/VM system consists of at least two virtual machines, the SERVER and at least one ADAPTER machine. The additional use of a HACC/VM monitor machine to control organizational processes is recommended. depicts a possible configuration scheme.

Figure 5: Example of a HACC/VM configuration

The term *TASK* and an appropriate control mechanism in HACC/VM have been introduced to enable simultaneous task processing. Furthermore, the interface communication between HACC/VM and connected AML systems has been moved to separate HACC/VM adapter machines.

The benefit of this concept is a more balanced distribution of the work load between several virtual machines without the necessity of developing complicated special dispatching algorithms. Furthermore, using a multiprocessor machine enables true parallel operations, which is not readily feasible when all functions are pooled in one CMS machine.

The communication within a HACC/VM system of the current version is performed by using the SMSG/VMCF protocol together with an interrupt control realized by the modul HACCWAIT.

For the AML/system a 3270 data stream protocol is used.

AML CONTROL COMPUTER

This is a system interface between the host and the AML robot system which is called the AMU (**AML M**anagement **U**nit) and is realized in the form of a PC.

In the current version HACC/VM is communicating with this inserted computer through a 3270 protocol, which requires, on the PC side, the use of a 3270 emulation or Token Ring card and the appropriate software. In principle, a host application is running on the PC/3270-monitor, which is operated by the PC as a *User*.

In this case it is irrelevant whether the PC is connected to the VM system locally (non SNA) or through SNA. A local (non SNA) connection is recommended solely because of better performance, independence from the SNA network and a higher stability of communication.

PREREQUISITES FOR THE USE OF HACC/VM

The HACC/VM system is intended for use under the operating systems VM/ESA or VSE/ESA.

It is also possible to use it under the VM operating system platform VM/XA SP 2.x. For this purpose the following requirements in the environment have to be met:

- CMS pipelines are a basic requirement (can be installed as a supplementory product)
- under VM/ESA, drive management by HACC/VM uses the CP command GIVE. Under certain conditions, restrictions may exist for the integration of particular tape management systems (e.g. DYNAM/T-CMS).
- APAR VM35846 and VM35847 or the new syntax of the CMS fullscreen commands have to be activated.

FUNCTIONAL DESCRIPTION

The HACC/VM System (**H**ost **AML C**ommunications **C**ontrol/VM) which can be operated under VM provides the following functions:

1. All Cartridge processing operations of tape management software run under VM or VM/VSE are recognized and the appropriate control data streams for the AML robot system are generated
2. Execution of administrative tasks, e.g. insertion and ejection of cartridges, scratch-tape management, cleaning of attached cartridge stations etc.
3. Automatic assignment of cartridge drives to the AML system (ATTACH / GIVE / DETACH)
4. Totally automatic and error tolerant operation as well as recovery options after system errors
5. Monitoring of robot functions
6. Priority control of tasks

In general a HACC/VM system is capable of making **all** decisions that can be made by a human operator of cartridge units and the cartridge archive on its own. The following functions are necessary to comply with specific criteria of the HACC/VM:

a) A filter mechanism has to capture all messages that are usually sent by subsystems (VSE guest machines with a HACC/VM supported tape management system, service machines of VM tape management systems etc.) to a console in the cartridge archive for processing by a human operator. The filter mechanism also has to initiate the appropriate actions for the AML system

   - In particular these are MOUNT instructions, from which at least the *VOLSER* number (or a TAPE pool identifier - e.g. SCRATCH) has to be determined for the tape to be mounted and possibly the address (real or virtual) for the cartridge station to be operated.
   - Furthermore, KEEP instructions (see topics: Scratch Tapes, Multi Volume Files) have to be recognized or automatically generated to release the used drives.

b) Actions regarding cartridge management, that are not initiated by a system message but are executed by a human operator, have to be processed automatically. Particularly this includes dismounting (KEEP) of cartridges which are discharged by the respective application without generating an appropriate message by the tape management system or the operating system.

c) A mixed operation must be ensured when cartridges are mounted manually on units which cannot be operated by the robot as well as automatically by the robot on units of the AML system.

- Based on the prompted *VOLSER* it is decided whether the cartridge is robot controlled

  or

- for mounting/dismounting requirements the robot has to execute the appropriate action on this drive for a specific unit address of the robot system

  or

- other organizational criteria, e.g. owner ID, vault (storage location), mode (density), pool ID (for cartridges and/or drives), determine whether a cartridge is to be mounted in a robot system or on manual units

d) To be able to mount cartridges which are not prompted by a *VOLSER* number, but by a tape pool identifier (particularly SCRATCH tapes) an adjustment is necessary between the catalogues of the respective subsystems of the tape management system and the scratch lists kept in the HACC/VM (which tapes are in which pool or which are scratch tapes?).

- Therefore it is necessary to maintain at least one scratch tape list within the HACC/VM system which has to be updated on a regular basis.

e) When the AML system architecture is asymmetrical it is not possible to mount each and every VOLSER on any cartridge unit.

- It is either necessary to maintain a special archive within the HACC/VM

  or

- for the HACC/VM to carry out special recovery operations in cooperation with the inserted PC of the AML system (AMU).

HACC/VM SYSTEM STRUCTURE

The HACC/VM system consists of the following components which are in fact service machines:

- Server
- Adapter
- Router
- Monitor

In its basic function the HACC/VM server is built as an automation tool to perform safely, trouble-free and without an operator.

The primary task of this server is to receive and appropriately transmit requests for mounting/dismounting cartridges. These requests are generated by and received from the subsystems (usually these are tape management systems under VSE or VM, but can be CMS machines or other server machines as well). The requests are transmitted to the AML robot system through the respective HACC/VM adapter.

The transmission of a request for mounting/dismounting a cartridge is performed either by a robot interface within the respective subsystem or by a “router“ function, where relevant messages of the subsystem are monitored.

The 3270 communication interface between the HACC/VM and the AML system as a further component of the HACC/VM package has been removed from the HACC/VM server environment in form of an adapter machine, because a virtual machine under VM/CMS works serially (up to VM/ESA), and therefore is supervised by the scheduling/dispatching of the VM operating system.

Thus, the HACC/VM - AML interface component is working quasi parallel (in case of multi processor machines even real parallel). The function responsible for this is the HACC/VM adapter machine. To reach the AML system the requests managed by the HACC/VM server have to be transferred to a HACC/VM adapter.

A major issue for automation systems like HACC/VM is to guarantee error-tolerant automatic operation. Thus, special care has to be taken regarding possible recovery interventions. Therefore the term *TASK* and an appropriate control mechanism were introduced in the HACC/VM. The requests channeled into HACC/VM are stored until the the AMU confirms the execution of the respective operation (positive/negative).

THE COMPONENTS OF HACC/VM

HACC/VM SERVER

STRUCTURE AND FUNCTION OF THE SERVER

The following scheme identifies the resources needed by the HACC/VM server machine for automated cartridge processing in a VM system connected to an AML robot system:

Figure 6: HACC/VM server resources

Explanation of Server Resources

Resource	Meaning
ALERT LOG	Here, all important events including abnormal termination of the server are recorded by the Return/Reason Code. Usually such information is also transmitted to an alert machine (alert box).
ARC	Archive catalogue • Volsers • Coordinates
CLEAN/CLEANEX	Manages the current (CLEAN) and the expired/ejected (CLEANEX) CLEAN VOLSERs within the HACC/VM system
LOG1/LOG2	Logging Dataset All events/processes occuring in a server, are recorded by the logging function. LOG1: Contains all incoming messages (e.g. mount requests of a VSE guest system) LOG2: Contains all outgoing messages (e.g. to a HACC/VM adapter machine)
MLOG	Message Log, saves all incoming messages of the CLIENT
PARM FILES	Description of all installation specific defaults (names, values, variables etc.) This information is used to reconstruct the internal system tables during a COLD start.
RUN Files	Backup copies of the system tables on the hard disk. This information is used to reconstruct the internal system tables during a WARM start or a RESTART.

Resource	Meaning
SCRATCH LIST	Contains the information about SCRATCH VOLSERs (picking list) provided by the tape management system. Theoretically, a separate SCRATCH LIST can be maintained for each CLIENT system. Within a list the opportunity exists to distinguish between SCRATCH pools; e.g. OWNER=
System Tables	All information necessary for processes of the server is managed by these internal tables.
TLOG	Task Log, contains all executable tasks (outbound)
TRACE LOG	Wrap around TRACE file for diagnostic purposes
UNIT LOG	Controls the cleaning procedure (use count etc.)

BASIC FUNCTIONS (MODULS) OF THE HACC/VM SERVER MACHINE

I. Interrupthandler
II. Inbound - Message - Processor
III. Scheduler
IV. Dispatcher
V. Task manager
VI. Outbound - Message - Processor
VII. Command - Exec - Processor
VIII. Alert processor

The following figure depicts the interactions of the basic functions of the HACC/VM server machine:

Figure 7: Flow diagram of HACC/VM server functions

MESSAGE MANAGEMENT

This server function acts as the central receiving and transmitting unit.

The messages/requests generated by the *clients* are buffered and transferred to task control. During the execution of a task the process is reversed, i.e. task control generates different types of messages on its own, e.g.:

- Routing
- Answering
- Monitoring

which are (re-)transmitted to one or several of the *clients*.

Message flow diagram

Figure 8: HACC/VM message flow

Message flow description (scheme):

1) "SUBSYSTEM" client

- Requests from an operating system (VSE guest system and/or CMS user) regarding Mount/Keep have to be transmitted to a HACC/VM server machine:

  Message (standard format):

  ```
  "M V=123456 U=680 J=XYZ4711 ..."
  ```

2) Server

- MLOG/TLOG task processing
- Dispatching/Scheduling
- Routing "MOUNT" to an adapter

3) "ADAPTER" client

- Integration of the request into the internal waiting queue of the HACC/VM adapter
- synchronization of a task via 3270 protocol in relation to an AML robot system.

4) AML communication and archive computer

- Read-out and transfer of a command for execution to the appropriate robot process computer.
- synchronization of answering (acknowlegement) to the adapter.

5) "ADAPTER" client

- synchronization of decision feedback tasks of the AML robot system via 3270 protocol and transmission to the HACC/VM server machine.

6) Server

- Handling of return information from the adapter according to the return code.

*) "MONITOR" client

- User interface for controlling and process observation of the HACC/VM components.

TASK CONTROL

Processing of messages/requests is supervised by Task Control. The processes initiated by Task Control are determined by the message type and the request contents.

MLOG Dispatch

- All tasks recorded in MLOG are processed. During this procedure the availability of resources necessary for a meaningful and successful completion of the task is investigated in relation to the AML system. If the resources are available, the SWAP procedure is performed.

- SWAP

  Here, the task is removed from MLOG (SWAP OUT) and incorporated as *executable* into TLOG (SWAP IN).

- Resources (availability check)

  - Unit (free)
  - VOLSER (not mounted/in archive)
  - Robot (active)
  - HACC/VM adapter or AMU communication

TLOG Dispatch

- All tasks located in TLOG are processed. Depending on the task status (Acknowledge) it is decided whether the task remains in the waiting queue or is to be transferred to the task manager for further processing.

- Task status

INIT	when tasks are transferred from MLOG to TLOG the respective task is labelled with the status INIT. As soon as the task was successfully transferred to the respective HACC/VM adapter, the task status is set to WAITADP.
END	when a task was processed successfully by the AML system or task processing was interrupted by HACC/VM with the command CANCEL, the task status is switched to END. During automatic recovery by HACC/VM the trouble causing task is terminated by HACC/VM (END), and a new task is generated.
WAITADP	the task was transferred to the respective HACC/VM adapter and is waiting for a response from the AML system (waiting for adapter).
WAITOPR	an error occured while processing a task which requires manual intervention by an operator (waiting for operator intervention).
WAITRCY	an error occured while processing a task. HACC/VM attempts to solve the problem automatically (waiting for recovery completion).

Closed loop control circuit: Inbound - Dispatch - Outbound

Figure 9: HACC/VM closed loop control circuit, Inbound-Dispatch-Outbound

Inbound: Example MOUNT - Request

Figure 10: HACC/VM Inbound-Mount-Request

Outbound: Example MOUNT - Request

Figure 11: HACC/VM Outbound-Mount-Request

HACC/VM ADAPTER

FUNCTIONAL PRINCIPLES OF THE ADAPTER

The basic function of the adapter machine results from the architecture with regard to the technical integration of the AML system and the host environment. The adapter machine is the target for all requests to the AML system.

The HOST/PC interface is addressed in parallel by two different program levels:

- On the host side the interface is operated by the HACC/VM adapter machine. The interface between Host and PC is realized logically (and to some extent also physically) as a 3270 monitor. All inputs and outputs of this monitor (or monitor sessions) are managed by the HACC/VM adapter machine.

- On the PC side the AML archive software of the AMU is simulating the operator of a 3270 host application, who is reading out the logical monitor of the HACC/VM adapter machine and is performing keyboard entries.

The communication protocol has the following levels:

1) Physical protocol level: IBM 3270 data stream
2) Logical protocol level: Send/Receive protocol for data exchange between HACC/VM and the AML robot system. The protocol uses a synchronization device to warrant data integrity of the messages.

A description of the logical protocol level can be found in the chapter *Logic of the 3270 Communication Between HACC/VM and AMU,* beginning on page 21.

The message flow (tasks, commands, return information etc.) between the different program levels occurs asynchronously as a duplex procedure.

Several adapters can be installed within a HACC/VM system. The requirement or necessity for this option is primarily dependent on the respective situation of the environment.

- On the host side an adapter always consists of a separate virtual machine which operates **one** logical 3270 monitor as an interface. This also means that for each adapter one 3270 emulation needs to be set up on an AMU.

- One PC with the AML archive software (AMU) can manage and control one robot system consisting of one or several robots (IC). The periphery of one robot system consists of the following components:
  - Cartridge device(s) IBM 3480/90 or compatible
  - Cartridge storage (rack/ linear storage device, towers)
  - Insertion/ejection unit

- Per adapter machine, a certain type information is defined and determined by parameter settings. Accordingly, a fixed assignment exists for:
  - specific unit addresses (34xx)
  - a particular robot system
  - pre-defined CLEAN cartridges

HACC/AMU INTEGRATION VIA 3270 PROTOCOL

Figure 12: HACC/VM robot integration via 3270 protocol

LOGIC OF THE 3270 COMMUNICATION BETWEEN HACC/VM AND AMU

The interface between HACC/VM and the AML system is based on a 3270 data stream protocol. A 3270 formatted monitor is used by the HACC/VM and the AML system for message exchange.

```
0 0 1  Id=VMXADPT1 HACC V1R1L1.0 20/02/92 18:40:29  VM/AML S=000009 R=000010 +
<A1H1A00,0010,MV  , ,    ,-,-,591,*11001,* ,00000201,00030101,20/180725,       >
 rrssttt,Sqnr,Comd,A,Retc,S,R,Dev,Volser,St,Param1..,Param2..,Timestamp,+++++++
<AAAAAAA,0010,MV  ,P,    ,1,1,591,*11001,  ,00030101,00000201,20/180530,       >
 ...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
 hc
                                          -
                                          -
                                          -
                                          -
                                          -
 ATTN 51 3522.868870 67229 Dsp=TermWait Prev.Read=ENTER:3/80:<AAAAAAA,0010,MV  ,

 VM/AML 3270 24*080
```

The different fields have the following interpretation:

Line_01: Head segment, beginning from the left it contains the following data groups.

(1) Send/Receive Synchronization

Field_1[1]: 0 = Archive-PC (AMU) Flag (unlocked)
1 = " " " " (locked)

Field_2: 0 = Host-Receive-Mode (receive ready)
1 = Host-Send-Mode (data send)

[1] Flag 1 is used to emulate the 3270 "keyboard locked" status. Prior to an I/O interrupt (ENTER) caused by the AMU this flag has to be set to 1 by the AMU. The 3270 session is accessible for writing by the AMU only when it is released by resetting this flag to 0 by HACC/VM.

Field_3: 0 = PC-Receive-Mode (receive ready)
1 = PC-Send-Mode (data send)

(2) Adapter-UserID/Archiv-Synonym-Code/Release

(3) Timestamp (last used at host)

(4) Text

(5) Send/Receive Counter

Line_02: Host transmission area:
- Host write-only
- PC read-only

Line_03: Layout mask for lines 2 and 4

Line_04: PC transmission area
- PC write-only
- Host read-only

Line_05: Column scale

Line_06: Adapter - command line

Use of the command line excludes the simultaneous use of line_04. Line_04 and line_06 exclude each other. In the following the range of commands is explained.

Line_07-21: Empty, not used

Line_22: Status

Line_23: Error messages

Line_24: "Environment data" for AML PC

SEND/RECEIVE SYNCHRONIZATION

The first 3 flags of the first line of the AMU 3270 monitor control the communication between the host (HACC/VM adapter) and the AML (AMU).

- Host transmits: data can be sent from the host to the AMU, if flag 2 is set to 0. When the data transfer in the host-transmission area is completed (line 2), flag 2 is set to 1 which tells the AMU that data is available in the host transmission area. Simultaneously, flag 1 is set to 0 to make the 3270 session accessible for the AMU.
- Host reads: as soon as an I/O interrupt (ENTER) generated by the AMU occurs, flag 3 is set to 1, and data is available in the PC

transmission area. To notify the AMU that data have been duly read-out, flag 3 is reset to 0. Simultaneously, flag 1 is set to 0 to make the the 3270 session accessible for the AMU.

- AMU transmits: As soon as flags 1 and 3 are reset to 0, the AMU has the opportunity to transfer data to the PC transmission area. When the data has been completely transferred to this area, flags 1 and 3 have to be set to 1 and transmission has to be initiated by triggering an I/O interrupt (ENTER).
- AMU reads: To notify the HACC/VM adapter that data sent by the host has been accepted by the AMU, flag 2 is set to 0 and simultaneously the 3270 session is made unaccessible for the AMU by setting flag 1 to 1. An I/O interrupt (ENTER) is triggered to have the HACC/VM read out the data. Only when this flag is reset to 0, the AMU is able to transmit more data to the host.

HACC/VM MONITOR

The monitor function is an optional component of a HACC/VM system, and is not necessarily required for ordinary operation.

However, there are reasons for the use of a monitor component. Under applied conditions, a number of requirements occur which necessitate the use of the monitor function.

These experiences have been taken into account during the development of the HACC/VM monitor and lead to the performance parameters listed below:

- The monitor function can be activated in each authorized VM machine. A tiered authorization is possible.

- Observation (recording) of the HACC/VM complex with regard to
  - Message flow from/to subsystem(s)
  - Message management (queueing, tasks, events, etc.) within the server environment
  - Message management (queueing, orders, etc.) within an adapter environment
  - Event dependent messages, e.g. infos, statistics,
  - Demand for resources, error situation(s), reply situation(s), etc.

- Debugging of HACC/VM components

- Start/Stop initiation of server/adapter/router components

- Operator Control

  - dynamic activation/de-activation of monitor machines (user)
  - dynamic modification of the authorization criteria of a monitor user
  - dynamic modification of the user interface

- HELP menus
- HOST command support, i. e. a temporary switch to the CP/CMS/EXEC environment is possible
- command processing
- query of all HACC/VM resources, e.g. log, task, stati, system tables, devices, etc.
- cancel/remove/resume/retry processing against active processes of the server/adapter environment in the area of logging/queueing/tasking, etc.
- emulation of subsystem processes
- robot command execution facility, i. e. any possible AML command can be executed interactively
- SET system parameters, dynamic modification of system wide parameters
- REFRESH system tables (parameters)
- MODIFY status of running tasks in the system

Detailed information about the operation of the monitor function can be obtained in the chapter *HACC/VM Monitor,* beginning on page 24 of this manual.

HACC/VM ROUTER

The router component of HACC/VM serves as an integrative interface for different subsystems which communicate via application messages on a respective console with the HACC/VM.

The HACC/VM router machine is implemented as a VM PROP (programmable operator) Machine. This means that application-relevant messages are defined in a routing table and processed through specific action routines.

Typical examples for using the HACC/VM routers are the integration of DYNAM/T-CMS or Dynam/B (see chapter *DYNAM/T-CMS and DYNAM/B* on page 33) and VM:Tape or VM:Backup (see chapter *VM:Tape and VM:Backup* on page 34) into the HACC/VM.

PROCESSING LOGIC

Central control of the HACC/VM system is performed by the server machine. As a rule all tasks are transmitted to the server machine by the CP command, SMSG (VMCF protocol).

In this chapter the processing logic is described for the following functions of the HACC/VM system:

- Mount
- Keep
- Drive management
- Drive cleaning
- Insertion and ejection
- Foreign mount processing

MOUNT

Function: To transport a cartridge from the archive and/or E/I/F area of the robot system to a drive and the insertion of the cartridge (mounting).

A mount task is generated either from a message of the tape management system used by an appropriate filter mechanism, or by direct transmission of the mount message to HACC/VM. In principle, the format of this mount message can be defined without restrictions, and therefore it can be easily adjusted to the respective environment. The implemented standard format of the mount message has the following structure:

$$\mathbf{M\ V} = \begin{Bmatrix} \textit{volser} \\ \text{SCRTCH} \\ \textit{poolid} \end{Bmatrix} \mathbf{U} = \begin{Bmatrix} \text{ANY} \\ \textit{cuu} \end{Bmatrix} [\mathbf{O} = \textit{owner}] [\mathbf{J} = \textit{jobname}]$$

When the HACC/VM server machine receives such a SMSG message from a valid subsystem via the VMCF interface, this message is transferred upon receipt as a Task into the MLOG.

The following procedures process a mount request within the HACC/VM server:

- Interrupt Control: The interrupt control verifies the sender of a message and generates a Task entry in the MLOG
- Scheduling Control: An available drive has to be selected to process a mount request by the robot. Furthermore, a certain volser has to be selected from the scratch list to process a scratch mount request. If no drives are available (e.g. because all are occupied by other mount requests), or the respective HACC/VM scratch list does not contain scratch volsers, the respective Task remains in the MLOG.

  As soon as all necessary resources are available during a later HACC/VM dispatch cycle the Task is removed from the MLOG and transferred to the TLOG.
- Dispatch Control: The mount request is transmitted to the respective HACC/VM adapter machine. When the AMU gives a positive feedback the task entry is removed from the TLOG.
- Recovery Control: In case of negative feedback (e.g. volser not in robot system) recovery operations are initiated if possible (and useful).

- In case of a scratch mount request another scratch volser is selected from the scratch list and the mount is attempted again.
- In case of a mount request for a specific volser which is not available in the robot system the mount request is suspended until the appropriate cartridge has been inserted into the robot system. Then, the respective task has to be re-started by the HACC/VM command RETRY.

The following flow diagram depicts the sequence of operations during a mount task:

Figure 13: Mount Logic

KEEP

Function: Removing an ejected cartridge from a drive and transporting it to the appropriate site in the archive or E/I/F area of the robot system (dismounting).

A prerequisite for sucessful dismounting (KEEP) by the AML robot system is that the cartridge must already be ejected (ejected from the drive).

In other words, the subsystem that transmits a Keep request to HACC/VM has to ensure that the respective drive has performed the ejection.

The dismounting (Keep) of a cartridge from a drive is performed depending on the assignment of this drive to an appropriate subsystem (c.f. ***Fehler! Verweisquelle konnte nicht gefunden werden.*** on page **Fehler! Textmarke nicht definiert.**). However in general the HACC/VM attempts to perform a dismount of the respective drive subsequent to a Keep request of a subsystem. This Keep request directed to the HACC/VM system is generated automatically by the respective SSCI interface.

An automatic Keep is also performed by the HACC/VM in the following situations:

- in case of dynamic drive assignment when the subsystem releases the drive using the CP command DETACH or logs it off from the system with LOGOFF.
- during the start of HACC/VM when a cartridge is in one of the drives controlled by HACC/VM.

In the following situation dismounting of the drive is **not** carried out automatically:

- when an attempt is made to cancel a manual drive assignment using the HACC/VM command RELEASE, as long as a cartridge is in the drive. The reserve status of the drive is cancelled by HACC/VM only if no cartridge is in the drive.

DRIVE MANAGEMENT

The assignment of a drive to a subsystem can be performed in the following ways:

- **dynamic** assignment via the CP command GIVE in the case of a mount requests by a subsystem. The mount request is submitted using a unit pool identifier (ANY). The drive assignment is realized by HACC/VM as soon as the respective mount is positively confirmed by the AMU and an appropriate verification of the mounted volume has been performed by HACC/VM.
- **manual** assignment via the HACC/VM command RESERVE. In this case the respective drive is excluded from automatic drive management by the HACC/VM until the drive is released by the HACC/VM command RELEASE. It is possible to assign a drive to a certain subsystem (ATTACH) using the HACC/VM command RESERVE or to put it into the status CP FREE (e.g. to relinquish drive management to a tape management system).
- **static** assignment by defining certain HACC/VM start parameters. The respective drives are not assigned by HACC/VM.

The manner in which drives are assigned to certain subsystems subsequent to mount requests is determined during the configuration of the HACC/VM system by the definition of the subsystems. Refer to *HACC/VM Installation & Customization Guide*.

In case of manual and static (dedicated) assignment of drives, all cartridge dismounts (keeps) of those drives are performed only upon request of the respective subsystem.

DRIVE CLEANING

The read/write heads of the cartridge drives become contaminated during operation. At certain intervals, depending on the type and manufacturer of the drive, it is necessary to mount a cleaning cartridge to reduce the probability of read/write errors.

These intervals are controlled by the microcode of the drive or by the control unit.

The drives or control units should be configured such that cartridges mounted by the AML robot system are accepted even when the limiting value is exceeded.

To comply with the cleaning intervals recommended by the manufacturer of the drive, HACC/VM performs a preventive cleaning of the AML robot system drives. In other words, each drive is cleaned at a regular interval by automatically mounting a cleaning cartridge. For this purpose HACC/VM carries an internal counter for each drive which is set for that particular drive type.

When, after a mount procedure, the internal drive counter exceeds the defined threshold value, a cleaning cartridge is mounted on this drive immediately after the dismounti (keep). Simultaneously, the HACC/VM internal counter is reset.

Due to the limited life span of a cleaning cartridge HACC/VM operates an additional counter specifically for the number of cleaning procedures per cleaning cartridge. When the maximum number of cleaning procedures is exceeded, the cleaning cartridge is automatically removed by HACC/VM (removal area E00) and should be replaced as soon as possible with a new one with the same barcode label.

The HACC/VM commands CLEANTAB and UNITTAB and the HACC/VM parameter file S05$VOLS are used for the administration of automatic drive cleaning by HACC/VM.

HACC/VM maintains the following internal files to manage the necessary counters for automatic robot drive cleaning.

- HACCLEAN NAMES — The number of cleaning procedures that can still be performed before the cleaning cartridge is automatically removed is recorded in this file. Drive cleaning is always performed with the first cleaning cartridge (for the respective drive) available in this file.
- HACCLEAN CLEANEX — As soon as the capacity of a cleaning cartridge is exhausted, the respective entry is removed from the file HACCLEAN NAMES and is recorded in the file HACCLEAN CLEANEX.

- HACUNIT NAMES — In this file, for each drive of the AML robot system, are recorded the total number of cartridge mounts performed as well as the number of mounts still possible until the next scheduled cleaning.

The HACC/VM system is to be notified of a new cleaning cartridge which replaces an exhausted one in the AML robot system by using the HACC/VM command CLEANTAB ADD.

INSERTION AND EJECTION

Function: Insertion and ejection are defined as the introduction or removal of cartridges to or from the AML robot system. The E/I/F area of the robot system is used for this purpose.

The space of the E/I/F area is partitioned into different fuctional areas defined by logical names during the AML configuration of the AMU. It is important to remember that one ejection area is defined as E00 and is needed by HACC/VM for the automatic ejection of exhausted cleaning cartridges and for damaged data carriers (label error).

Cartridges can be inserted or ejected using special HACC/VM commands. The slots of the insertion/ejection area are managed exclusively by the AMU in cooperation with HACC/VM.

INSERTION

Function: During insertion the cartridges are put into a defined insertion area of the E/I/F area by human operators. From there the robot moves them to slots of the archive area.

After the cartridges selected for insertion are placed in a defined insertion area of the E/I/F area, insertion by the robot system can be started using the following HACC/VM command via the HACC/VM monitor machine:

ROB *sys,rob* IN I*nn*

where the parameter I*nn* is the identification of the insertion area defined on the AMU as well as in HACC/VM.

Now the AML robot system moves all cartridges located in the defined insertion area to appropriate slots of the robot archive area. The cartridges are identified through a barcode label that must be on each cartridge of the robot system.

If the barcode label of a cartridge selected for insertion is unreadable[1] the cartridge is moved to a “special problem“ area of the E/I/F area, and an appropriate message is delivered.

[1] Barcode-label damaged or non existent; wrong placement of cartridge into insertion area (cartridge flipped).

The coordinate of the slot where the inserted cartridge is placed depends on the archive management of the robot system:

1. in case of hierarchical storage a fixed coordinate is assigned to each volser during AMU configuration. The slot is always used for the same cartridge. In other words as long as a cartridge is ejected the slot for this cartridge remains empty.
2. in case of dynamic storage the slot coordinate of a volser can be released when the cartridge is ejected[1], which makes the slot available for a cartridge inserted later.

For each inserted cartridge the message HACADM512 is displayed which contains the volser and the cartridge's slot coordinate. A description of this message can be found in the chapter *AML robot commands of the HACC/VM Operator Guide*.

To obtain a protocol file when a larger number of cartridges are inserted it is recommended to use the HACC/VM Batch_Command_Facility.

Note that the respective Batch_Command file does not contain any other robot commands except the insertion command.

Beispiel

A cartridge has been inserted in the insertion area defined as I01.

The insertion is initiated by the HACC/VM command

ROB 1,1 IN I01

All cartridges located in the I01 insertion area are moved to respective slots in the system's archive area by robot #1 of robot system #1.

EJECTION

Function: During ejection, cartridges are transported by the robot from slots in the archive area to a defined ejection area of the E/I/F area from which they can be removed by operations personnel.

[1] Using the AML command EJT (EJect Total)

To eject a cartridge from the archive area of the robot system, the following HACC/VM command can be executed at a HACC/VM monitor machine:

ROB *sys,rob* EJ E*nn volser*

or in case of dynamic archive management by the command:

ROB *sys,rob* EJT E*nn volser*

where the parameter *volser* defines the cartridge selected for ejection and E*nn* represents the designation of the ejection area of the E/I/F area defined on the AMU as well as in HACC/VM.

The robot removes the cartridge selected for ejection from the respective slot of the archive area and transports it to the next available slot of the designated ejection area. When this area is filled, further cartridges can be ejected only after operations personnel have emptied this area.

Depending on the archive management of the robot system the following types of ejection are distinguished:

1. in case of hierarchical storage a fixed coordinate has been assigned to each volser during the configuration of the AMU. The ejection can be performed only with AML EJ, the slot continues to be reserved for the respective volser and cannot be used for other cartridges.
2. in case of dynamic storage the ejection can be performed with the AML command EJT (Eject Total). Then, the slot coordinate is available for another cartridge which was inserted later.

Beispiel

Using the HACC/VM command

ROB 1,1 EJ E01 004711

the cartridge with volser 004711 is put into the first available slot of the ejection area defined as E01 of robot #1 in robot system #1.

Special exits are provided by HACC/VM for insertion as well as ejection of cartridges of the AML robot system. Using these exits, storage slot labels of tape management systems can be updated dynamically for instance. For a detailed description of these exits refer to *Utilities and Exits,* beginning on page 1.

The **Batch_Command_Facility** of HACC/VM is recommended particularly for the ejection of large numbers of cartridges (cf. chapter *Batch Processing (Batch-_Command_Facility)*, p. 3).

Using this facility, files can be sent to the HACC/VM server machine for initiating the ejection of several cartridges. It is possible to eject a number of cartridges exceeding the capacity of the ejection area of the robot system by this means.

When the ejection area is filled, so that no more cartridges can be ejected, the control task which was generated by HACC/VM for a Batch_Command_File is switched to a disconnected mode. When the ejection area is emptied by an operator, this task can be re-activated by the HACC/VM command

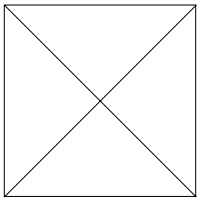
Modify **B**atch **RES**ume *SpoolId*

and the ejection procedure is continued. Here, *SpoolId* is the spool file number of the respective read file.

FOREIGN MOUNT

Cartridges that have no bar code label at all, or one which is not defined in the AML robot system, can be mounted in the foreign mount area. For cartridges to be mounted in this area on drives of the AML robot system no bar code identification is conducted by the vision system of the robot. Also, HACC/VM does not perform a label verification.

The area consists of a number of defined slots with coordinates in the insertion/ejection area of the robot system with specifically assigned logical volsers of the AML system. These logical volsers of the AML robot system have the following structure:



Furthermore, any volser can be dynamically assigned to these foreign mount slots via HACC/VM. A mount request for such a dynamically assigned volser is realized by the HACC/VM system. The volume defined by the respective coordinate of the foreign mount area is mounted by the robot system.

Beispiel

Five slots of the insertion/ejection area were defined as a foreign mount area in the AML robot system. The HACC/VM command

M FMSD 1 CONNV FREMD1

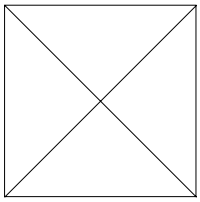
assigns the logical volser FREMD1 to the first slot of this area. During a later mount request for the volume labelled with the volser identification FREMD1, the cartridge of the respective slot (here: slot 1) of the foreign mount area is mounted. A Keep request returns the cartridge to this foreign mount slot.

Cartridges can be mounted directly[1] from the foreign mount area of the E/I/F area of the robot system, when, in the mount request of the subsystem, the logical AMU volser of the respective slot is stated.

[1] i.e. conversion of a logical HACC/VM volser into the respective logical AMU volser

Beispiel

A volume located in the 2nd slot of the foreign mount area F01 of the AML system 1 is mounted in response to the mount request of a subsystem defined in HACC/VM.



The foreign mount area of an AML robot system can be used simultaneously by several connected host systems as long as simultaneous access to a slot in this area is prevented by organizational measures.

HACC/VM STARTING PROCEDURES

Requirements for the operation of an AML system under VM are:

- operability of the AML system taking into consideration the respective AML starting procedure (incl. AMU) - refer to the appropriate AML manual
- the HACC/VM server machine
- the HACC/VM adapter machine necessary for communication with the system
- a functioning 3270 connection for communication between the HACC/VM adapter machine and the AMU
- the necessary interface components, e.g. HACC/VM router, HACC/VSE etc.

Normally HACC/VM should be started automatically at the same time as the IPL of the VM system. During the installation of HACC/VM the starting procedure HACCSTRT EXEC was generated. Using this procedure the HACC/VM server machine can be started by a separate virtual machine. It is presumed that the virtual machine possesses the CP command category (A) for executing the CP command XAUTOLOG. Usually, PROFILE EXEC is not installed on minidisk 191 of the HACC/VM server machine.

By executing the HACC/VM starting procedure HACCSTRT EXEC in the AUTOLOG1 machine it is possible to start HACC/VM automatically at VM IPL.

The HACC/VM adapter machine which is responsible for communication with the AML system is started automatically by the HACC/VM server, if the appropriate definition (ADPS=(SVR,...)) is specified in the parameter file S03$AUSR.

HACC/VM server and HACC/VM adapter also can be started by an authorized CMS user via the HACC/VM user interface (HACC/VM monitor) (chapter *HACC/VM Monitor*, page 24). For this purpose the CMS user needs the respective CP command category (A) for starting the HACC/VM server machine.

A re-start of HACC/VM by a HACC/VM monitor machine may be necessary when the configuration of the HACC/VM software is changed.

Interruption of the server function always results in a loss of control of the HACC/VM component regarding its operative environment.

Whereas the HACC/VM functions are re-startable within themselves, an interrupted communication between the subsystems, e.g. VSE and the HACC/VM server, require appropriate interventions for recovery.

For instance, such a situation occurs when the subsystems generate mount requests during an interrupted communication (subsystem - HACC/VM server), because the server component of HACC/VM is unable to accept the requests under such conditions. After re-starting the HACC/VM system all open requests of the subsystems can be finished manually via the HACC/VM command EMU.

USER INTERFACE

Usually, HACC/VM is controlled and operated by the HACC/VM monitor function. A HACC/VM monitor machine is a CMS machine defined as a HACC/VM operator. The monitor function is activated by calling the procedure:

HACMON

(access -read only- to the HACC/VM product minidisk 192). If the ACCESS of the HACC/VM product minidisk 192 and the call of this command are specified in the PROFILE EXEC of the respective CMS machine, the HACC/VM monitor function is activated automatically during user LOGON.

This function realizes all logical/organizational requirements necessary for running and operating the HACC/VM system environment.

The principles and functions of the monitor component offer a variety of applications.

In the following, the main characteristics of the monitor user interface are explained:

The main menu of the user interface has the following segments:

(1)	Header segment	(title and status lines)
(2)	Data segment	(dynamic area)
(3)	Footer segment	(PF key layout)

Depending on the operation, different sub-menus (windows) are activated within the data segment:

STANDARD:	HACC/VM "Message Flow" Window. Line-by-line display in Scrolling Mode (top/down).
ATTENTION:	Window for command entry
RDEVICE:	Window for Real Device Activity Reflection.
HELP:	Window for Help function.

All menus (windows) are generated and processed using the CMS "virtual screen" window technique.

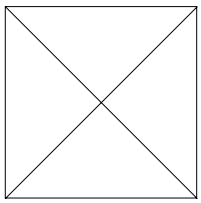


Figure 14: HACC/VM monitor

In the command line all HACC/VM commands can be dispatched, for which the respective CMS machine has authorization (see parameter description S02$OUSR and S17$CAUT in *HACC/VM Installation & Customization Guide for details*).

SHUTTING DOWN HACC/VM

SHUT-DOWN OF THE HACC/VM SERVER

As a rule, a controlled shut-down of the HACC/VM server function, and subsequently, the function of the complete HACC/VM system is initiated by an authorized HACC/VM monitor machine using the HACC/VM command SHUTDOWN.

SHUTDOWN SVR ONLY only the HACC/VM server machine is shut-down, all other HACC/VM components remain active.

SHUTDOWN SVR * the complete HACC/VM system is deactivated

SHUT-DOWN OF A HACC/VM ADAPTER

The controlled shut-down of a HACC/VM adapter function is regulated by a HACC/VM monitor machine authorized for that purpose:

SHUTDOWN SVR * LOGOFF Useful, when a central shut-down of the whole HACC/VM system is intended. In this case all HACC/VM components are deactivated.

SHUTDOWN ADP ì *adp* ü í ý î *sys,rob* þ Using this method a defined HACC/VM adapter machine is shut-down. The HACC/VM server machine remains active, incoming requests are suspended until the respective adapter function is restored.

Application: Maintenance of the respective robot system.

SHUT-DOWN OF A HACC/VM ROUTER

The controlled shut-down of a HACC/VM router function can be performed as follows:

SHUTDOWN SVR * — Useful, when a central shut-down of the whole HACC/VM system is intended. In this case all HACC/VM components are deactivated.

SHUTDOWN ROUTER *router* — Using this method a defined HACC/VM router machine is shut-down. The HACC/VM server machine remains active. Operations of all subsystems integrated through the respective HACC/VM router regarding the robot system are interrupted.

SHUT-DOWN OF A HACC/VM-MONITOR SESSION

The shut-down of a HACC/VM monitor function is entirely uncritical and has no functional effects on the HACC/VM system.

The controlled shut-down of a HACC/VM monitor function can be performed using the monitor command QUIT.

SPOOL FILE PROCESSING

The HACC/VM system is able to process special requests that were transferred as files into the virtual reader of the HACC/VM server machine.

The following procedures are distinguished:

- AML robot commands are transmitted to HACC/VM in a file for execution as a *BAT[1] instruction. Such a file is called a Batch_Command_File (*Batch Processing (Batch_Command_Facility)*, page 3).
- Output lists for special applications (e.g. the storage area list of the tape management system) are transmitted to HACC/VM for further processing. This procedure is called Batch_Preprocessing (*Batch Preprocessing*, page 9).
- To enable scratch mounts, HACC/VM needs information from the tape management system about the scratch cartridges. Supply and processing of the respective scratch list of a tape management system is called the scratch facility (*Scratch List* , page 11).

HACC/VM processes only Batch_Command_Files that are transferred into the virtual reader of the HACC/VM server machine using the correct CP spool category (cf. description of the parameter SERVER_SRDR of the parameter file S08$SERV in *HACC/VM Installation & Customization Guide*).

Also, only one Batch_Command_File per subsystem or HACC/VM administrator is processed at a time.

The following figure depicts the processing of a file that is read-in and processed via the virtual reader of the HACC/VM server machine:

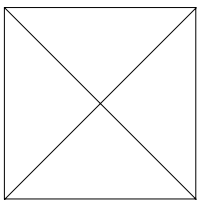


Figure 15: Spoolfile processing by HACC/VM

[1] the syntax of a *BAT instruction corresponds to the syntax of the HACC/VM command ROB

Spool File Processing

BATCH PROCESSING (BATCH_COMMAND_FACILITY)

The function called *Batch_Command_Facility* allows the transmission of several AML commands (as *BAT instructions) for execution to HACC/VM in a *Batch_Command_File*.

Usually Batch_Command_Files are generated by subsystems and transferred for processing via VM spooling into the virtual reader of the HACC/VM server machine.

The concept of this function also considers the requirements of the computing facilities during the execution of certain procedures within the automatic tape operation archive, e.g.:

- insertion[1]
- ejection
- inventory

All authorized AML commands (defined in parameter file S10$ROBC) can be processed using the "Batch_Command_Facility". In practice, this application will be confined mainly to the procedures mentioned above.

- MOunt and KEep commands are **not** permitted in a Batch_Command_File.
- *BAT commands of only **one** AML system are permitted in a Batch_Command_File.
- Batch_Command_Files used during insertions may contain only a single *BAT IN statement.

The principle of this device is based on the following aspects:

(1) All batch commands of a certain category are summarized in one list.

[1] Insertion by a Batch_Command_File is useful to obtain the response-list of the respective batch processing as a protocol.

Normally, such a list is machine-generated by appropriate administrative programs and transmitted to the HACC/VM server for further processing (cf. section *Batch Preprocessing*, beginning on page 9).

(2) HACC/VM processes a Batch_Command_File as a background task. I.e., online requests have priority over commands of a Batch_Command_File.

The individual commands of the Batch_Command_File are processed serially. The results (responses of the AML system) are recorded in a response list generated by HACC/VM.

(3) After processing a Batch_Command_File the result (response list) is re-transmitted to the sender.

(4) The functional loop of batch processing is closed by this re-transmission (acknowledgement).

Here, revision of the response list of a batch command process can be started automatically.

FUNCTIONAL DESCRIPTION

The authorizations and definitions necessary for executing the "Batch_Command_Facility" for the respective subsystems must be recorded in the parameter file S04$SUBS.

Receipt and processing of a Batch_Command_File and the generation of a response list (acknowledgement) is performed according to these definitions:

(1) Receipt of a new Batch_Command_File

An administrative user, defined for HACC/VM or a subsystem, transfers the Batch_Command_File to the HACC/VM server via VM spooling using the respective CP spool category.

A valid Batch_Command_File contains one *BAT statement per line. The syntax of this particular HACC/VM batch is described in the section *Command Structure: *BAT* beginning on page 7.

The file name of a Batch_Command_File is BTCHCMD[1]. In case of other file names, HACC/VM examines, whether a Batch_Preprocessing routine exists for the situation (see *Batch Preprocessing* beginning on page 9) or whether it is the name of a defined scratch list (see *Scratch List* beginning on page 11).

HACC/VM accepts a Batch_Command_File from VM spool and saves it for further processing under the following file names on the A-Minidisk of the HACC/VM server machine:

FN	UserId of the sender
FT	*nnnn*CMD (*nnnn* is the Spool-ID from which the Batch_Command_File was received)

(2) Processing a Batch_Command_File

- A batch control task is generated in TLOG
- Each batch command (*BAT statement) in the Batch_Command_File is received as an individual task (Batch Subtask in MLOG) and at the same time it is removed from the Batch Command_File. The maximum number of

[1] In case of VSE subsystems a Batch_Preprocessing Routine has to be defined for the Power-Jobname. Cf. Section *Batch Preprocessing* beginning on page 9.

Batch_Commands that can be transferred at once into MLOG is defined in the parameter file S08$SERV.

- A response list with the following names is generated on the A-Minidisk of the HACC/VM server machine:

  FN UserId of the sender

  FT *nnnn*CMD (*nnnn* is the Spool-ID from which the Batch_Command_File was received)

- After processing all batch subtasks, the respective batch_controltask is completed.
- Depending on certain subsystem definitions (S04$SUBS) the generated response list is sent to the subsystem or to the HACC/VM administrative user (SERVER_ADMU in S08$SERV).

(3) Process observation of *BAT tasks

- *BAT tasks are handled by HACC/VM like ordinary subsystem requests. The only difference is that these requests have lower priority and can be acknowledged in form of a response list. This response list is transferred to the virtual reader of the respective subsystem or, under certain conditions, to the first defined HACC/VM administrative user under the following names:

  FN UserId of the sender

  FT *nnnn*REC (*nnnn* = Spool-ID)

The following figure depicts the sequence of a batch processing:

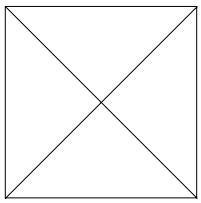


Figure 16: Processing of Batch_Command_Files

COMMAND STRUCTURE: *BAT COMMAND (BATCH_COMMAND)

The *BAT comand is the functional equivalent of the HACC/VM command ROBCOMM.

The *BAT command permits the execution of the AML commands defined in the HACC/VM parameter file S10$ROBC.

The command can be executed only with an authorized Batch_Command_File.

Command	Operands (position 1-n)
`*BAT:adapter|(sys,rob):cmnd:template`	

The *BAT command and the operands are separated from each other by a colon (**:**).

A list of AML commands and the necessary operands can be found in the section on the ROBCOMM command in the HACC/VM Operator Guide.

Operand 1:

adapter	UserId of the respective HACC/VM adapter machine
(*sys*,*rob*)	System-Id and robot-Id of the respective AML system and robot. In a Batch_Command_File *BAT commands of only **one** AML system are permitted.
cmnd	AML command[1] defined in the HACC/VM parameter file S10$ROBC. The AML robot commands MO (Mount) and KE (Keep) are not permitted.
template	the parameters of an AML command separated by a colon (**:**)

[1] When the AML robot command IN (Insert) is used in a Batch_Command_File to obtain a protocol file of an insertion started with this command, no other AML command is allowed in this Batch_Command_File.

Beispiel

The HACC/VM administrative user HACMNT sends the CMS File BTCHCMD INSERT with the entry:

```
*BAT:HACADP1:IN:I01
```

to the HACC/VM server machine.

All cartridges located in the insertion area I01 are to be inserted into the AML system, to which the HACC/VM is communicating via the HACC/VM adapter machine HACADP1.

Upon completion of the insertion, the user HACMNT receives a response list HACMNT *spid*REC were an entry exists for each cartridge inserted (*spid* = SpoolId).

Beispiel

The user CMS1, defined in HACC/VM as a CMS subsystem, sends the CMS file BTCHCMD EJECT with the following entries to the HACC/VM server machine:

```
*BAT:(1,1):EJ:E01:I00300
*BAT:(1,1):EJ:E01:I00301
*BAT:(1,1):EJ:E01:I00302
*BAT:(1,1):EJ:E01:I00303
*BAT:(1,1):EJ:E01:I00304
```

The cartridge with the volser numbers I00300-I00304 are to be removed from AML System 1 (robot 1) and placed in the ejection area E01.

After processing this Batch_Command_File, user CMS1 receives a response list CMS1 *spid*REC where the AML system's response to each *BAT command can be found.

BATCH PREPROCESSING

The term Batch_Preprocessing is used for a process that enables the automatic transformation of a list that is in the virtual reader of the HACC/VM server machine into a Batch_Command_File using the Batch_Preprocessor routines.

Batch_Command_Files created in this way are further processed by HACC/VM with the procedure *Batch Processing (Batch_Command_Facility)* which is described on page 3.

A possible application is the initiation of an automatic ejection process from a storage slot list generated by a tape management system.

These Batch_Preprocessing routines are called by HACC/VM and must return at least one numerical return code to HACC/VM[1]. Here, the call REXX by HACC/VM has the following format:

PreProcRc = Batch_PreProc(*infile*,*outfile*)

The respective Batch_Preprocessing routine has to generate a file with the name *outfile* (fn ft fm) from a file called *infile* (fn ft fm) which contains the original form of the list. This new file contains only the *BAT statements (*Batch Processing (Batch_Command_Facility)* page 3).

The assignment of a Batch_List[2] name to a corresponding Batch_Preprocessing routine is performed by the HACC/VM parameter file S21$BPRE (see also *HACC/VM Installation & Customization Guide*).

The HACC/VM system contains two examples of Batch_Preprocessing routines that can be used as templates for developing own applications.

The routine HACMVDYN EXEC (and XEDIT) processes a movement list of the tape management system DYNAM/T.

The example HACEJDYN EXEC (and XEDIT) generates a Batch_Command_File for automatic ejection of cartridges from a DYNAM/T pull list.

---

[1] A string of the following format has to be returned by the Batch_Preprocessing routine to the calling level:

Returncode [error message]

If the return-code is unequal to 0 the transmitted error message is displayed by HACC/VM.

[2] The file name must not be BTCHCMD, because this is reserved for Batch_Command_Files which already contain *BAT statements that can be processed by HACC/VM.

SCRATCH LIST PROCESSING

PRINCIPLE

The basic principle of the *Scratch_List_Facility* function is to support the operation of different tape management systems.

The request for mounting empty tapes (scratch tapes) by the tape management system can be presented in the following ways:

1. the tape management system selects the volser of an empty cartridge from its own catalogue and requests a mount message for mounting this specific cartridge. This form of requesting an empty cartridge is called "**specific request**" and for the HACC/VM neither relevant nor recognizable.
2. the selection of an empty cartridge is not performed by the tape management system. It is requested to mount a cartridge from a specific scratch pool in a mount message. This form of requesting an empty cartridge is called an "**unspecific request**“ or "**scratch mount**". During this procedure the tape management system often performs the assignment of the drive where the scratch cartridge is to be mounted (AVR - automatic volume recognition).

In the following, the automatic selection of a volser is described complying with a scratch mount by HACC/VM (scratch substitution).

Usually the request for mounting an empty cartridge (unspecific request) comes from a mount message for a cartridge of a specific scratch pool. In principle several scratch pools can exist which need to have an unique identifier (scratch pool Id).

A scratch pool is a certain amount of free volumnes which are identified and listed by the tape management system (usually by a scratch run).

Such a list is needed in case of manual operation e.g. for insertion and supply purposes.

In case of automatic operation using the AML robot system and HACC/VM this organizational tool is needed for scratch tape processing.

FUNCTIONAL DESCRIPTION

If the volser demanded in a HACC/VM mount request is identical with the label defined in a HACC/VM scratch pool (parameter file S13$SCRA) a specific volser is selected and mounted from the respective internal scratch list.

To enable this procedure, HACC/VM must always be supplied with valid scratch lists of the respective tape management system.

HACC/VM generates internal scratch lists from scratch lists of the tape management system that can be used for substitution purposes during scratch mounts. The transfer of these lists is performed via the virtual reader of the HACC/VM server machine similarly to the Batch_Command_Facility (*Batch* Processing (Batch_Command_Facility), page3).

The necessary definitions have to be made in the HACC/VM parameter files S04$SUBS and S12$SCRL.

The following figure depicts scratch list processing in HACC/VM:

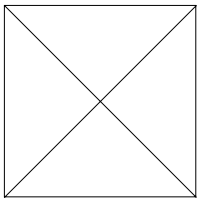


Figure 17: Scratch_List_Facility

(1) Transfer of a new scratch list

A HACC/VM administrative user or the respective subsystem transfers the scratch list of the appropriate tape management system with the correct CP spool category to the virtual reader of the HACC/VM server machine.

(2) Scratch list maintenance

If the file name of the spool file in the HACC/VM parameter file S12$SCRL is defined as a scratch list, the appropriate Scratch_List_Processing routine is called.

These Scratch_Processing routines are called by HACC/VM and have to return at least one numerical return code to HACC/VM[1]. Here, the call REXX of HACC/VM has the following format:

ScrProcRc = Scratch_Proc(*infile*,*outfile*)

The respective Scratch_Processing routine has to generate a file called *outfile* (fn ft fm) from the file called *infile* (fn ft fm) which contains the original list. This newly generated file contains an entry of the following format for every scratch cartridge that can be mounted by HACC/VM:

Column	Cell Description	
1-6	*Volser*	the volser of the scratch cartridge
8-15	*OwnerId*	if no OwnerId is available, this cell has to be filled with the one-digit hexadecimal value x'04'

From this file, HACC/VM generates an internal HACC/VM scratch list with the following structure:

Column	Cell description	
1-6	*Volser*	the volser of the scratch cartridge
8-15	*OwnerId*	if no OwnerId is available, this cell has to be filled with the one-digit hexadecimal value x'04'
17	*SysId*	the designation of the AML Systems (1 or 2)

[1] A string of the following format must be returned by the Scratch_Processing routine to the calling level:

Returncode [error message]

If the return-code is not 0 the transmitted error message is displayed by HACC/VM.

Column	Cell description	
19-22	*RobId*	a binary designation of the robot which can be employed for mounting the cartridge. The following values are possible: 0001 robot 1 0010 robot 2 0011 robot 1 + 2
from 24	comment	

(3) Processing of a scratch tape request

For each MOUNT request, HACC/VM determines whether it is a scratch request. Scratch pool designations are defined in the HACC/VM parameter file S13$SCRA.

In this case, HACC/VM determines a specific "*volser*" from the respective scratch list and replaces it in the MOUNT request which is then transmitted to the AML system.

This "*volser*" is removed from the scratch list and entered into the Used_Scratch_List. This list contains an entry of the following format for each scratch cartridge used by HACC/VM:

Column	Cell description	
1-6	Volser	Volser of the scratch cartridge used by HACC/VM
8-15	OwnerId	the OwnerId of the scratch cartridge used (x'04' for *empty* OwnerIds)
17	yyyy.ddd	Date when the cartridge was used by HACC/VM
26	hh:mm:ss	time when the cartridge was used by HACC/VM

If the preset minimum number of scratch tapes within one list drops below the limit, HACC/VM takes steps to request new scratch lists.

Because the current version of HACC/VM does not maintain an archive of the cartridges located in the AML system the following problem exists:

If the tape management system does not conduct a regular inventory of storage slots (e.g. DYNAM/T), the number of scratch cartridges in the AML robot system may not be determined exactly. However, the exits INSEXIT and EJCTEXIT provided by HACC/VM offer inventory management.

UTILITIES AND EXITS

EXITS

The HACC/VM system provides the following exits for use and adjustment:

- INSERT (INSEXIT EXEC)
- EJECT (EJCTEXIT EXEC)

These exits have the format of REXX procedures and are called by HACC/VM when the respective files (INSEXIT EXEC or EJCTEXIT EXEC) are found on a minidisk of the HACC/VM server machine.

INSERT EXIT

If a HACC/VM exit procedure with the name INSEXIT EXEC is found on a minidisk of a HACC/VM server machine, it is called up for each cartridge inserted by the AML system.

The respective volser, system-Id and robot-Id of the inserting robot are transmitted as parameters. A sample exit (INSEXIT BVSEXEC) can be found on the HACC/VM product minidisk. This sample exit can be used to change the respective storage slot label in the tape management catalogue of the volser just inserted, when the tape management system BVS is used.

Note that re-starting (cold start) the VM system may cause the loss of spool files, when a BVS is used and an insert exit is applied to update the storage slot labels of the inserted cartridges in the BVS catalogue by transmitting those spool files to the BVS server machine.

EJECT EXIT

If a HACC/VM exit procedure with the name EJCTEXIT EXEC is found on a minidisk of the HACC/VM server machine, this procedure is called-up for each cartridge ejected by an AML system.

In this case the respective volser, system Id and robot Id of the ejecting robot are transmitted as parameters.

UTILITIES

HACUT001 (INVENTORY)

A Batch_Command_File can be generated using the program HACUT001 to perform an inventory (logical/physical) on a certain range of coordinates.

HACUT001 [**T**n] [*fromseg* [*toseg*]] é ì **ULC** ü ù
ê í **INC** ý ú
ë î þ û

Tn	Tower number (Default T1)
fromseg	Starting segment (1-32)
toseg	End segment (1-32)
ULC	Logical inventory - inquiry of the AMU data base (default)
INC	Physical inventory - robot action

The Batch_Command_File generated is automatically sent to the HACC/VM server machine.

The Userid for the utility HACUT001 has to be valid for the HACC/VM Batch_Command_Facility.

HACUT002 (READ-IN OF BATCH__RECEIPT_FILES)

All Batch_Receipt_Files located in the virtual reader can be read-in using the program HACUT002.

HACUT002

Each Batch_Receipt_File is saved on hard disk as a CMS file with the file name BTCHREC *nnnn* A. (*nnnn* is the SpoolId of the ReaderFiles read-in)

HACUT003 (PROCESSING OF BATCH_RECEIPT_FILES)

Using the program HACUT003, the Batch_Receipt_Files read-in with HACUT002 can be prepared for further processing with HACUT004.

HACUT003 *fn ft fm*

Here, *fn ft fm* designates a Batch_Receipt_File read-in with the utility HACUT002.

If this Batch_Receipt_File contains responses to INC commands the files INC-VOLS BTCH*nnnn* and INC-COOR BTCH*nnnn* are generated. If this Batch_Receipt_File contains responses to ULC commands the files ULC-VOLS BTCH*nnnn* and ULC-COOR BTCH*nnnn* are generated.

HACUT004 (MERGE OF INVENTORY DATA)

Using the program HACUT004, the already existing files INC-VOLS ALL, INC-COOR ALL, ULC-VOLS ALL and ULC-COOR ALL can be updated by the files generated with HACUT003.

HACUT004 *fn* **BTCH***nnnn*

The input files INC-VOL BTCH*nnnn*, etc. are deleted after being processed.

HACUT005 (DISPLAY A SEGMENT NUMBER OF AN ALL-FILE)

Using the program HACUT005, the number of all segments can be displayed for which inventory information is available in the respective file INC-VOLS ALL etc.

HACUT005 *fn ft*

HACUT006 (DISPLAY SEGMENTS BY DATE)

Using the program HACUT006, the date of the last inventory is displayed for each segment, for which inventory information is available in the respective file INC-VOLS ALL etc.

HACUT006 *fn ft*

INTEGRATION OF TAPE MANAGEMENT SYSTEMS

Currently the following types of integration exist to support tape management systems:

- Support of the HACC/VM system **integrated** into the tape management system. The tape management system BVS belongs to this type of integration.

- **direct** integration via appropriate interface programs into the respective environment. This is used for instance for the integration of VSE guest systems operated under VM which are not equipped with a tape management system with integrated support of a ADIC/GRAU robot system. Specific exit programs in the respective tape management system inform HACC/VM (via SMSG) about relevant messages (MOUNT etc.) or events (OPEN, CLOSE) in the VSE system.

- **indirect** integration via a router machine. For this type of integration a HACC/VM router machine is connected to an appropriate service machine of the tape management system (via SCIF or as a special operator machine of the service machine) in a way that the HACC/VM and the tape management system can communicate. This method of integration is used for various tape management systems that are operated directly under VM, e.g. DYNAM/T-CMS (DYNAM/B) and VM:Tape (VM:Backup).

HACC/VM supports the following tape management systems:

- under VM
  - DYNAM/T and DYNAM/B (- Computer Associates)
  - VM:Tape and VM:Backup (- Sterling)
  - BVS (Tape management system [=**B**and **V**erwaltungs **S**ystem] - Infosoft)
  - ADSM (**A**DSTAR **D**istributed **S**torage **M**anager - IBM)
  - HACC/BR (**B**ackup/**R**estore - ADIC/GRAU)

- VSE under VM
  - BVS (Tape management system [=**B**and **V**erwaltungs **S**ystem] - Infosoft)
  - DYNAM/T (Computer Associates)
  - EPIC (Legent/CA)

VSE TAPE MANAGEMENT SYSTEMS

HACC/VM supported VSE tape management systems are categorized into those with direct integration using the component HACC/VSE and those with an integrated interface to HACC/VM or the ADIC/GRAU robot system. Indirect integrations as applied for the different VM tape management systems are not used for the VSE guest systems.

TAPE MANAGEMENT SYSTEMS WITH INTEGRATED SUPPORT

BVS/VSE

Infosoft's tape management system BVS is equipped with an integrated interface to HACC/VM. Therefore, the HACC/VSE[1] component is **not** needed for integrating a VSE guest system into a BVS tape management system. When BVS is used in a VSE guest system, the BVS component BVR is applied, which transmits Mount and Keep requests via a CP SMSG command to the HACC/VM system for cartridges located in the AML system.

Figure 17*Figure 18: Configuration when using* BVS depicts the integration of BVS into the ADIC/GRAU robot system:

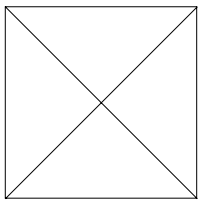


Figure 18: Configuration when using BVS tape management

In case of data carriers of the BVS tape management catalogue, BVS identifies the cartridges located in the ADIC/GRAU robot system according to the storage slot label (Vault). The storage slot lists generated by BVS are used by HACC/VM for automatic ejection (see also chapter *Insertion and Ejection* beginning on page 34). For the purpose of insertion a specific insertion exit can be used by HACC/VM (cf. section *INSERT Exit* , page 1), which updates the storage slot label in the BVS tape management catalogue for inserted cartridges.

[1] HACC/VSE is used only for those VSE tape management systems which have no integrated support from the GRAU robot system.

DIRECT INTEGRATION USING HACC/VSE

For using the tape management systems DYNAM/T and EPIC/VSE, some components of the software HACC/VSE are necessary (under certain conditions EPIC/VSE can be integrated without HACC/VSE, cf. chapter *EPIC/VSE,* beginning on page 28).

The basic software of HACC/VSE consists of the following VSE phases:

- ❒ $ROBEXIT — Interface between VSE guest system and HACC/VM
- ❒ $JOBEXnn — VSE JCL exit routine for dismounting drives with an incorrect ejection
- ❒ $HACCVSE[1] — Command-interface and Recovery controller program
- ❒ HACCPVSE — HACC/VSE parameter phase to be assembled from the source HACCPVSE ASSEMBLE which needs to be adjusted

For DYNAM/T the following HACC/VSE interface programs are needed in addition:

- ❒ DYNEXIT — DYNAM/T interface controller program
- ❒ DYNEX55
- ❒ DYNEX60
- ❒ DYNEX60P

THE COMPONENTS OF HACC/VSE

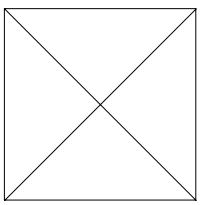


Figure 19: Illustration of HACC/VSE components

[1] When the component $HACCVSE is used in a separate partition, the installation of HACC/VSE $JOBEXITs is not necessary.

HACCPVSE

Phase HACCPVSE contains all the parameters for controlling HACC/VSE. Furthermore, HACCPVSE provides the memory areas commonly used by all HACC/VSE components.

The parameters are defined in the source file HACCPVSE ASSEMBLE. After conversion under CMS using the HACCASM command the VSE phase HACCPVSE has to be generated from the respective TEXT Deck (Link Option SVAPFIX)[1].

The phase HACCPVSE has to be loaded into the SVAPFIX area to activate HACC/VSE.

[1] During assembling with HACCASM a Job is automatically generated which can be used for the installation of parameter phase HACCPVSE.

$ROBEXIT

The HACC/VSE component $ROBEXIT serves as a communication interface between the VSE guest system and HACC/VM. The CP command SMSG is used as protocol here.

Mount requests for the robot system and status information is transmitted to HACC/VM using SMSG.

This is a one-way communication:

HACC/VSE ($ROBEXIT) ® HACC/VM (Server)

i.e. HACC/VM does not return information to HACC/VSE.

$JOBEXIT

Cartridges that were mounted on a robot drive because of an earlier mount request are returned by HACC/VM when one of the following requirements is fulfilled:

- the VSE guest system has ejected the drive and directs a keep request to HACC/VM (by the interface between tape management and HACC via $ROBEXIT)
- the robot drive is released by the VSE guest system (DETACH)

When a tape data set that has been processed on a robot drive was not closed (CLOSE) properly, usually no Keep request is sent to HACC/VM. In other words, the drive remains occupied even after job end or job step limits and is not available for further mounts.

In the following situations HACC/VSE $JOBEXIT releases all robot drives used by a job as soon as these are no longer assigned (ASSIGN) by VSE:

- after discontinuation of a job or CANCEL
- after step change
- after job end

Starting with VSE/ESA, the activation of the HACC/VSE component $HACCVSE is recommended in a dynamic VSE partition. Besides other features $HACCVSE performs background monitoring of all robot drives that are assigned to a VSE guest system. When $HACCVSE is used HACC/VSE $JOBEXITs are no longer necessary.

The following table summarizes arguments for and against the use of indivdual components which can be used for decision making:

	exclusively $JOBEXIT	exclusively $HACCVSE	$JOBEXIT and $HACCVSE
Control	event driven	timer driven	event and timer
Overhead	not relevant	minor	minor
Command Interface	not relevant	available	available
Installation	complicated	easy	complicated
Resource	• SVA • does not occupy partition	• occupies dynamic or static partition	• SVA • occupies dynamic or static partition

Table 1: Advantages and disadvantages of $JOBEXIT versus $HACCVSE

The HACC/VSE $JOBEXIT can be integrated into a VSE guest system as follows:

- as the sole VSE $JOBEXIT. The HACC/VSE $JOBEXIT is installed as $JOBEXIT.PHASE in the VSE system library IJSYSRS.SYSLIB instead of the dummy $JOBEXIT provided by IBM. The HACC/VSE $JOBEXIT can also be installed in a different VSE library, if precautions are taken so that it is loaded into the SVA when the HACC/VSE is started.
- additional customized $JOBEXIT. In this case the customized $JOBEXIT has to be renamed and HACC/VSE has to be configured so that the renamed $JOBEXIT is called by the HACC/VSE $JOBEXIT.
- under VSE/ESA the opportunity exists to use several $JOBEXITs. In this case the HACC/VSE $JOBEXIT has to be installed as $JOBEX*nn*.PHASE (*nn* - 00 to 09) and has to be entered into the $JOBEXIT list called $JOBEXIT ASSEMBLE. After conversion (and link) this list has to be provided $JOBEXIT.PHASE on the VSE library IJSYSRS.SYSLIB.

$HACCVSE

The HACC/VSE component $HACCVSE is available for the following functions:

- Background observation of all robot drives assigned to the VSE guest system (ATTACH).

The access (ATTACHED) for the VSE system to robot drives which are not ASSIGNED is verified at regular intervals. When such a drive is found, some time is granted to the tape management system for assigning the drive to a partition. If, after this period of time, the drive has not been assigned to a partition, it is returned to HACC/VM (DETACH).

The HACC/VM automatically removes cartridges which may still be mounted from this drive and releases it for further mount requests.

- when a DYNAM/T tape management system is integrated, HACC/VSE maintains an internal table (Mount_Request Queue) of all mount requests that were transmitted to HACC/VM. As soon as an OPEN is executed toward the respective cartridge (by an application program), the request is removed from this table.

$HACCVSE checks this Mount_Request Queue at regular intervals and repeats mount requests that have exceeded a defined waiting period.

- $HACCVSE provides a command interface; in this situation, it is possible to display a Mount_Request Queue for instance. The operator communication with $HACCVSE can be activated by using the VSE command **MSG *pid*** (here, *pid* is the partition-Id where $HACCVSE is executed). The following commands are available:

QUEUE	display of the Mount_Request Queue
CLEAR	deletion of all entries of the Mount_Request Queue
ACT	dynamic activation of HACC/VSE
DEACT	temporary (dynamic) deactivation of HACC/VSE
STOP	termination of the HACC/VSE task $HACCVSE
CANCEL	manual deletion of entries of the Mount_Request Queue

HACC/VSE MESSAGES

HACC/VSE messages have the following general format:

HAC*XnnnC Text*

The output of HACC/VSE messages can be controlled by the parameter MSGLVL in the HACC/VSE parameter phase HACCPVSE. The suffix of the message header has the following meaning:

E	Error	(MSGLVL=1)
W	Warning	(MSGLVL=2)
I	Information	(MSGLVL=3)
D	Debug	(MSGLVL=4)

MESSAGES OF THE HACC/VSE COMPONENT $ROBEXIT

HACC000D F=*f* Unit=*cuu* Volser=*volser* Job=*jobname*

The SVA phase $ROBEXIT was called with the function *f*.

f	M	mount request
	K	keep (dismount) request
	U	update (information)
cuu	ANY	mount on an (directly) accessible drive
	addr	specific drive address
volser	SCRTCH	request of a scratch volume
	poolid	request of a volume of the pool *poolid*
	volser	request of a specific volser

HACC001D SMSG *hacsvr message*

A message has been transmitted to the HACC/VM server machine using the CP command SMSG.

HACC002D Detach wait entered...

HACC/VSE has detected a drive (ATTACHED), which at that time is not ASSIGNED to a partition. HACC/VSE waits for a period of time defined in HACCPVSE (DETWAIT), before it DETACHes the drive.

This allows a tape management system equipped with AVR (automatic volume recognition) to assign the drive to a partition.

HACC003I Keep ignored

A keep request for a manual drive is ignored. This may happen when drive addresses have been assigned in the paramter phase HACCPVSE which are not compatible with the VSE device definitions (ASI).

HACC006E Volume *volser1* mounted on unit *cuu* - KEEP for volume *volser2* ignored

A dismount request for cartridge *volser2* of drive *cuu* is ignored because, according to the internal drive table, the cartridge *volser1* is mounted on this drive.

HACC007E Error in ASSIGN Macro - R15=*hhhhhhhh*

The error *hhhhhh* occured during an attempted temporary assignment of a drive by HACC/VSE.

HACC008W Unit information overwritten for unit *cuu*

When the information was compared between the tape management system and HACC/VSE an out of date drive table was detected.

HACC009W Entry for Volser *volser* not found in volume chain

Removal of an entry of the internal Mount_Request Queue was attempted because of an OPEN by the tape management system or a comparable VSE console message. No entry was found, because either a second OPEN was executed toward an already mounted cartridge or Stealing (see *Glossay* page 4) occurred.

The message can be used for problem analysis.

HACC010E Error sending Request - R15=*hhhhhhhh*

HACC011E Phase probably not PFIXed

One or more components of the HACC/VSE system were not loaded properly into the SVA.

HACC012D Job delayed because of detach processing for unit *cuu*

Upon job end HACC/VSE checks all unassigned drives and may release them. In this case, HACC/VSE waits for a certain time period (defined in HACCPVSE) to allow the assignment of a drive to this partition by a tape management system equipped with AVR. When the drive is not assigned within this time frame it is released using the CP command DETACH.

HACC013E FRE/GETVIS error. HACC request ignored

An insufficient amount of GETVIS memory was defined in a partition.

HACC014E Abend routine of $ROBEXIT entered (Code *c*)

A serious error occurred in the HACC/VSE $ROBEXIT phase. Further processing may be impaired.

HACC015E Error x"*hhhhhhhh*" issuing SMSG command.

An error occured during VMCF communication with HACC/VM (*hhhhhhh* designates the CP return code)

HACC016E Invalid function call. Request ignored

The HACC/VSE $ROBEXIT phase was called with an invalid function code. This may indicate that incompatible components of HACC/VSE were installed.

HACC017E Internal error - stack corrupted.

A serious internal error of the HACC/VSE system occurred which may result in significant functional impairments of the tape management system.

ADIC/GRAU Storage Systems must be informed of the occurrence of this error.

HACC018D Unit *cuu* sensed by $ROBEXIT

HACC/VSE examines whether a cartridge is mounted on a non-assigned drive.

HACC019D Unit *cuu* unloaded by $ROBEXIT

HACC/VSE has dismounted a non-assigned drive and automatically generates a dismount (Keep) request to HACC/VM.

HACC020D Unit *cuu* temporarily assigned by $ROBEXIT

HACC/VSE examines non-assigned drives by temporary assignment for subsequent *Sensing*.

HACC021D Unit *cuu* unassigned by $ROBEXIT

The temporary ASSIGN by HACC/VSE has been cancelled again.

HACC022I HACC/VSE connected to HACC/VM server *hacsvr*

HACC/VSE has connected to the HACC/VM basis system. Appropriate mount and dismount (Keep) requests are now processed automatically by the robot system controlled by HACC/VM.

HACC023I HACC/VSE disconnected from *hacsvr*

Because of a communication problem with HACC/VM (e.g. HACC/VM server machine *hacsvr* stopped) the connection was terminated.

All mount messages of the tape management system have to be operated manually until the message HACC022I indicates that the connection to HACC/VM is re-established.

HACC024D Unit *cuu* detached by $ROBEXIT

The drive with the virtual address *cuu* was automatically released by HACC/VSE.

HACC025I $JOBEXnn version *version* level *level* Entry=*hhhhhhhh* activated

Displays the version and the SVA loading address of the HACC/VSE JCL exit $JOBEXnn.

HACC026I $ROBEXIT version *version* level *level* Entry=*hhhhhhhh* activated

Displays the version and the SVA loading address of the HACC/VSE program $ROBEXIT.

HACC027I *TMS name* TMS exit *phasename* version *version* level *level* activated.

Displays which version of the TMS interface program was activated by HACC/VSE.

The display of a message generated by HACC/VSE can be controlled via parameter MSGLVL of the HACC/VSE parameter phase HACCPVSE (see *HACC/VM Installation & Customization Guide*).

MESSAGES OF THE HACC/VSE COMPONENT $HACCVSE

HAC$01I HACC/VSE control task started. Loaded at x'xxxxxxxx'

The HACC/VSE Component $HACCVSE was started.The loading address is displayed.

HAC$02I HACC/VSE control task ended.

The HACC/VSE component was terminated using the command STOP.

HAC$03E HACC/VSE control task abnormally ended.

The HACC/VSE component $HACCVSE was terminated due to a serious error. This error must be referred to the service manager in charge.

HAC$04W scratch mount for job *jobname* (PID=*pid*) already repeated

The mount job waiting in the Mount_Request Queue has already been transmitted by $HACCVSE to HACC/VM. One must examine why HACC/VM did not mount a scratch cartridge (possibly scratch list empty, alle drives occupied).

HAC$05W specific mount for volser *volser* job *jobname* (PID=*pid*) still waiting ...

The displayed job is still waiting for a mount of the specific volume requested. If the respective cartridge is not present in the robot system, it either has to be inserted, to be available to the AML system via the foreign mount area, or be mounted on a manual drive and assigned to the VSE machine (using ATTACH).

HAC$06I scratch mount for job *jobname* (PID=*pid*) repeated

After expiration of the time period defined in the HACC/VSE parameter phase HACCPVSE (MAXWAIT) the displayed job is still waiting for the assignment of a scratch cartridge. The mount request is transmitted to HACC/VM again.

HAC$07I HACC/VSE mount queue control task started

A task has been started for background monitoring of mount requests generated by HACC/VSE for DYNAM/T.

HAC$08I HACC/VSE unit control task started

A task has been started for background monitoring of drives of the robot system.

HAC$09W Invalid HACC/VSE command

An invalid HACC/VSE command was entered.

HAC$10I HACC/VSE mount request queue empty

No entry was found during an inquiry of the Mount_Request _Queue using the HACC/VSE command QUEUE. The Mount_Request Queue is used only when the tape management system DYNAM/T is employed.

HAC$11I ID VOLSER CUU JOBNAME PID OWN DATASETNAME

Headline for the entries of the Mount_Request Queue displayed by using the HACC/VSE command QUEUE.

HAC$12I *id volser cuu jobname pid own datasetname*

Display of the entry of a Mount_Request Queue which is displayed by the HACC/VSE command QUEUE.

id	a unique ID number, which was assigned to the mount request by HACC/VSE
volser	SCRATCH or the specific volser to be mounted
cuu	ANY or the specific address of a drive to be mounted
jobname	the name of the job by which the mount request was generated
pid	the ID of the VSE partition in which the respective job is processed
own	the DYNAM/T OwnerId required by the mount message
datasetname	name of the DYNAM/T data set for which a cartridge is to be mounted

HAC$13E FREEVIS error occurred

A serious internal processing error occured. This error must be referred to the service manager in charge.

HAC$14E Invalid HACC/VSE command entered.

An invalid HACC/VSE command was entered.

HAC$15E GETVIS error occurred. UNIT task canceled.

A serious internal processing error occurred. The task for background monitoring of a robot drive was cancelled. This error must be referred to the service manager in charge.

HAC$16E GETVIS error occurred. QUEUE task canceled.

A serious internal processing error occurred. The task for background monitoring of the Mount_Request Queue for DYNAM/T was canceled. This error must be referred to the service manager in charge.

HAC$17I *n* mount requests have been deleted

The indicated number of entries of the Mount_Request Queue was deleted. For the respective jobs no recovery will be performed for the remaining mount requests.

HAC$18W HACC/VSE already active

Using the HACC/VSE command ACT an attempt was made to activate HACC/VSE although HACC/VSE was already started.

HAC$18I HACC/VSE activated

HACC/VSE was started with the HACC/VSE command ACT. Starting at this point, HACC/VSE attempts to transmit all mount or keep requests to HACC/VM. Also, background monitoring of the robot drives is interrupted.

HAC$19W HACC/VSE already inactive

Using the HACC/VSE command DEACT, an attempt was made to deactivate HACC/VSE even though HACC/VSE was not started.

HAC$19I HACC/VSE deactivated

HACC/VSE was deactivated using the HACC/VSE command DEACT. mount or keep requests are no longer transmitted to HACC/VM by the robot system.

HAC$20I Mount request *id* has been deleted from queue

The entry of the ID number *id* was removed from the Mount_Request Queue using the HACC/VSE command CANCEL.

HAC$21W Invalid mount id specified

For the HACC/VSE command CANCEL an invalid ID number was entered as a parameter. Valid ID numbers are 1-9999 or * to remove all entries from the Mount_Request Queue.

HAC$22W Mount request *id* not found in queue

For the ID number entered as a parameter in the HACC/VSE command CANCEL no entry was found in the Mount_Request Queue.

DYNAM/T-VSE

FUNCTION OF DYNAM/T

The tape management system DYNAM/T by the company Computer Associates can be used under VSE (DYNAM/T-VSE) as well as directly under VM (DYNAM/T-CMS and DYNAM/B). Particularly, the combination of both provides a lot of flexibility. All OPEN activities for magnetic tapes are recognized by DYNAM/T within the environment of the VSE operating system. Subsequently, a MOUNT message is transmitted to the respective monitor which usually induces the tape operator to mount the requested volume on a drive. In connection with DYNAM/T which can be employed under VM it is even possible to automate the assignment of the units by VM.

As a rule, DYNAM/T leaves it up to the tape operator to select a cartridge station for mounting a requested specific (or Scratch) cartridge or tape.

Generally, DYNAM/T permanently monitors the units controlled by DYNAM/T to determine whether the requested cartridges were already mounted (AVR - **A**utomatic **V**olume **R**ecognition). When a cartridge request is met within a certain time frame defined in DYNAM/T, no further intervention by the tape operator is necessary. The operator has to respond to a REPLY only after this time has elapsed.

Under VSE, the HACC/VSE component which was described earlier (see chapter *Direct Integration Using HACC/VSE,* beginning on page 4) is used to integrate DYNAM/T.

Since it is intended that the HACC/VM AML system functions as a quasi-human tape operator, the MESSAGE Exit (MSGHOOK), provided by C.A., was selected as the interface between DYNAM/T and HACC/VM. Furthermore, important information needed to guarantee the automation is transmitted to HACC/VM at the time of OPEN (OPNHOOK) and CLOSE (CLSHOOK).

The interface between HACC/VSE and HACC/VM is implemented using the exit program DYNEXIT and other version dependent[1] routines (DYNEXnn).

[1] Currently the version-dependent interface routines are DYNEX55, DYNEX60 u. DYNEX60P. Which routine should be used depends on the layout of the DYNAM/T interface block DTEXIT, rather than on the version of the DYNAM/T software. When the parameter TMS=(DYN,SELF) is defined, the DTEXIT block is analyzed automatically by HACC/VSE and the appropriate interface routine is used.

The following figure demonstrates the transmission of a DYNAM/T mount request to HACC/VM:

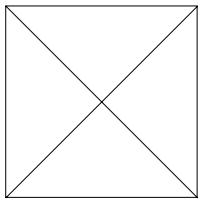


Figure 2: Functional diagram of HACC/VM and DYNAM/T-VSE

All relevant DYNAM/T console messages necessary for the control of automatic cartridge processing are assessed in the exit programs.

In this context the following tasks have to be fulfilled:

- decide whether a mount request to HACC/VM has to be transmitted for further processing (manually or automatically)
- recognize whether cartridge units of the AML system need to be ejected and transmit an appropriate dismount request (Keep) to HACC/VM.

After the successful transmission of the mount message, the original C.A. mount messages are modified by HACC/VSE to notify the VSE console that the request is being processed by HACC/VM.

HACC/VM INTERFACE TO DYNAM/T

The concept for integrating the DYNAM/T tape management system under VSE into a robot is presented here. It requires that the respective VSE systems are operated as guest systems under VM. In each of these VSE systems the HACC/VSE component and appropriate exit programs are needed:

- DYNEXIT (DYNAM/T MSGHOOK, OPNHOOK u. CLSHOOK)
- DYNEXnn (DYNEX55, DYNEX60, DYNEX60P)
- $JOBEXIT / $HACCVSE[1]
- $ROBEXIT
- HACCPVSE (HACC/VM parameter)

To enable HACC/VM to mount DYNAM/T scratch cartridges, it is necessary to synchronize the scratch set between DYNAM/T-VSE and HACC/VM. The appropriate REXX routine for processing DYNAM/T scratch lists can be found on the HACC/VM Product Minidisk (HACSLDYN XEDIT).

Using the proper definitions in the HACC/VM parameter files (S04$SUBS and S12$SCRL), synchronization of the HACC/VM scratch list can be automated.

DYNAM/T integration with HACC/VM takes into consideration (as far as possible) that automatic AML cartridge units and manually operated units are operated in parallel.

To warrant a parallel operation of manual drives and automatic robot drives the decision must be made possible, based on criteria to be defined, whether a job should to be transmitted to the robot system because of a DYNAM/T mount message.

The following criteria, defined in the HACC/VSE parameter phase HACCPVSE (see *HACC/VM Installation & Customization Guide*), can be used:

- **Drive address** (DEVPOOL=...)

  When a cartridge is requested on a certain drive,

  - CADT008A MOUNT SCRATCH *dtf* ON *cuu* DSN=*dsn* .....

[1] The Installation of HACC/VSE $JOBEXITs may not be necessary when the component $HACCVSE is used in a separate partition. See *Table 1: Advantages and disadvantages of $JOBEXIT versus $HACCVSE* on page 8.

the mount request is transmitted to HACC/VM only when the required unit address is defined in the parameter DEVPOOL.

- **Volser** (TAPPOOL=...)

For cartridges requested by specificing the volser

- CADT004A MOUNT VOLUME *volser dsn* SYS*nnn* JOB=*job*

the mount request is transmitted to HACC/VM only if the required volser is defined in the parameter TAPPOOL.

- **OwnerId** (OWNCHK=...)

If the DYNAM/T mount message contains information about an OwnerId (OWNER=),

- CADT008A MOUNT SCRATCH *dtf* SYS*nnn* DSN=*dsn* OWNER=*xx*

the mount request is transmitted to HACC/VM only if the required OwnerId is defined in the parameter OWNCHK .

- **Density** (MODE=)

If the DYNAM/T mount message contains information about the recording density (MODE= bzw. DEN=),

- CADT008A MOUNT SCRATCH *dtf* SYS*nnn* DSN=*dsn* MODE=xx

the mount request is transmitted to HACC/VM only if the required recording density is defined in the parameter MODE.

If no selection criteria are defined in the HACC/VSE parameter phase HACCPVSE, a mount request for the HACC/VM system is generated from each DYNAM/T mount message that contains the statement MODE=CART. I.e., it is implicitly assumed that all cartridges to be processed are located in the robot system.

Using appropriate definitions in the DYNAM/T tape management catalogue or by a specific instruction in the TLBL JCL statement controls, whether a mount request will be transmitted to the robot system (via HACC/VM) or whether it is a data carrier for manual processing.

AUTOMATIC MOUNT REQUESTS

The HACC DYNAM/T message exit DYNEXIT (MSGHOOK=DYNEXIT in the DYNAMT Macro) examines all relevant DYNAM/T mount messages. From these an appropriate mount request is generated for HACC/VM, as far as is permitted by the selection criteria.

The following table (Table 2) demonstrates the different types of mount requests generated and transmitted to HACC/VM using the CP SMSG command:

Header	Message Type	Mount Request to HACC/VM
CADT004x	MOUNT *volser* for Input	M V=volser U=ANY ID=nnnn D=...
CADT008x	MOUNT SCRATCH for Output	M V=SCRTCH U=ANY ID=nnnn ... M V=SCRTCH U=cuu ID=nnnn ...
CADT020x	MOUNT *volser* for multifile output	M V=volser U=ANY ID=nnnn D=...
CADT065x	MOUNT *volser* for output (ROTATE)	M V=volser U=ANY ID=nnnn D=...

Table 2: MOUNT requests from DYNAM/T

When DYNAM/T is integrated into the AML robot system via HACC/VM, the **AVR** (**A**utomatic **V**olume **R**ecognition) device must be activated, because HACC/VM does **not** respond to an awaited DYNAM/T *Reply*. An awaited DYNAM/T *Reply* is an exceptional situation for HACC/VM which cannot be fixed automatically.

Therefore, it is recommended that the STOP parameter of the DYNAMT configuration macro be set to a value that provides sufficient time for the AML - HACC/VM complex to mount a requested cartridge before DYNAM/T generates a new mount message that requires a *Reply*.

When resources are missing (e.g. when no drive is currently available, or the scratch list necessary for a mount request is empty) it may happen that one of the requested cartridges is not mounted for a longer period of time. However, this mount request has **not** been *forgotten* by HACC/VM (for uninterrupted operation).

When a scratch cartridge selected by HACC/VM (using the respective scratch list) and mounted by the AML robot is not accepted by DYNAM/T, this is not an error situation for DYNAM/T. DYNAM/T continues to expect the mounting of a proper cartridge.

Because HACC/VM does not receive information from DYNAM/T about a "*quasi*" rejected SCRATCH cartridge, HACC/VM examines, prior to assigning a drive, whether the mounted cartridge may be file protected (as a rule, cartridges of the AML system should **not** be file protected), and whether the label is identical to the expected label (label check).

If it is determined that the mounted cartridge is useless for DYNAM/T, the cartridge is automatically ejected and a new SCRATCH cartridge is mounted.

If a cartridge examined by HACC/VM is not accepted by DYNAM/T-VSE anyway, the Recovery function of the HACC/VSE component $HACCVSE is activated. The drive occupied by the cartridge which is not used by DYNAM/T is released and the respective mount request is sent to HACC/VM again. See section *$HACCVSE,* beginning on page 9 of *The Components of HACC/VSE* for further information.

AUTOMATIC DISMOUNT/KEEP

In contrast to mount requests sent to the robot system, dismounting the cartridges after processing by an application program is more complex. If the dismount/keep request sent to HACC/VM does not exist for a unit which has been busy by a previous mount request, it usually remains busy from the perspective of HACC/VM, and therefore is not available for further mount requests. To avoid this situation dismount/keep requests are generated on different levels of DYNAM/T, HACC/VM, and AML.

	Description	KEEP trigger	KEEP request
1	The application that caused MOUNT also generates the respective KEEP	DYNAM/T Close Exit (CLSHOOK=DYNEXIT) The regular case for a correctly performed job	K V=*volser* U=*cua* J=*jobname* O=*owner* P=*pid* D=*dsn*
2	At the end of a JOB step the cartridge units are checked and for stations where the unit is already dismounted, a KEEP is generated	$JOBEXIT when the 2nd and all following // EXEC Control command occur	K V=*volser* U=*cua* J=*jobname* O=*owner* P=*pid* D=*dsn* In the worst case scenario for a unit already dismounted at step end, the respective KEEP is generated only at job end
3	At JOB end usually all used cartridge units are dismounted and a CLEANUP command is sent to HACC/VM	$JOBEXIT at /& at the end of a jobstream	K V=****** U=ALL J=******** O=** P=*pid* D=
4	As soon as HACC/VM detects that a KEEP request is missing	E.g. when a MOUNT requestor for a unit which, from the perspective of HACC/VM is busy, is no longer "logged on" (or is DISCONNECTed).	internal

	Description	KEEP trigger	KEEP request
5	At JOB end those drives of the AML system are released (DETACH) automatically, which are currently assigned to the specific VSE system (ATTACHED) and not used by another partition	$JOBEXIT at /& at the end of a jobstream	internal

Table 3: KEEP requests of VSE and DYNAM/T

At job end, at the latest, HACC/VM is informed to dismount all stations which were loaded with cartridges by the AML system for the respective VSE partition.

EPIC/VSE

To integrate the tape management system EPIC/VSE the use of the system automation software **FAQS/ASO** is presumed. Besides other features, this software allows for the response to certain messages of the VSE system with automatic execution of pre-defined action routines.

For integration into the robot system the following REXX procedures (IMODs) are used:

- HACEP001 Mount request (EP00n MOUNT TAPE.....)
- HACEP088 Keep request (EP088 DISMOUNT CART....)
- HACEP047 e.g. for LIBR backup jobs (EP047 ...)
- HACEP164 for automatic cartridge initializing
- HACM065 for processing special HACC/VM messages
- CLEANCUU alternative to HACC/VSE

These IMODs are delivered together with the HACC software and must be installed properly.

Use of the HACC/VSE component is recommended instead of installing the CLEANCUU IMOD. See section *$HACCVSE* of chapter *The Components of HACC/VSE,* beginning on page 9*.*

To enable HACC/VM to mount EPIC scratch cartridges, a synchronization of the scratch set between EPIC/VSE and HACC/VM is necessary. The proper REXX routine for processing EPIC/VSE scratch lists can be found on the HACC/VM product minidisk (HACSLEPI XEDIT).

Furthermore, the ejection of data sets generated by the AML system can be automated using the vault movement lists.

When EPIC/VSE is integrated into an AML robot system using HACC/VM, the AVR device (**A**utomatic **V**olume **R**ecognition) of EPIC has to be inactivated. HACC/VM responds to upcoming prompts from the tape management system.

The following issues need to be considered when operating drives manually and by the AML robot system in parallel:

- the tape pool (TSIDPOL and IMODINST EXEC) is used as a distinctive character for SCRATCH processing. Only scratch mount requests of the robot pool defined as tape pools are transmitted to the AML system for further processing.

- for specific mount requests (particularly insert processing), pre-defined volser ranges (IMODINST EXEC) determine whether the mount request is to be processed by the AML system.

The mount messages EP001 or EP002 generated by EPIC/VSE elicit the call of the respective HACC IMOD (MOUNTVOL or MOUNTSCR) by FAQS/ASO. Depending on the parameters (volser range, tape pool, drive addresses etc.) defined during installation (IMODINST EXEC) the respective mount task is transmitted to HACC/VM.

After a drive with the requested volser was ATTACHED by HACC/VM this drive is ASSIGNed by the AVR function of EPIC/VSE to the respective application (partition).

During a CLOSE executed by the application, EPIC/VSE ejects the drive and generates the message EP088. Due to this message the HACC IMOD DISMOUNT is called by FAQS/ASO which releases (DETACHED) the drive (if it is a drive of the AML system). Subsequently, HACC/VM dismounts (KEEP) the drive automatically which is then again available for further applications.

The following schematic depicts the events during EPIC/VSE tape processing in connection with HACC/VM and an AML robot system:

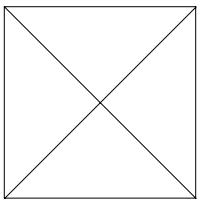


Figure 21: EPIC/VSE scratch mount flow diagram

Certain programs do not terminate a tape job properly. This includes - besides interrupted jobs (due to operator intervention or programm error) - the IBM utility LIBR.

In the special case of the program LIBR, the HACC IMOD HACEP047 is called when the message EP047 occurs, which automatically releases the used drive (DETACH).

In other situations, the following options for automation exist when drives requested by HACC/VM are not released:

- When using HACC/VSE, during each step change or each job end it is examined[1], whether drives, not needed by the AML system, are ATTACHED to the VSE system. Then, these drives are released automatically.
- The examination and release of drives by the HACC IMOD CLEANCUU can be performed either on a time-controlled basis via FAQS/ASO or as a result of certain VSE console messages (e.g. EOJ). However, a sometimes undesired side effect occurs in that CLEANCUU displays a large number of messages on the VSE console (output of the EPIC/VSE command EP STATUS).

[1] When the component $HACCVSE is activated in a separate partition the robot drives are examined at regular intervals and may enventually be released. In this case too, the use of HACC/VSE $JOBEXITs is not necessary.

VM TAPE MANAGEMENT SYSTEMS

TAPE MANAGEMENT SYSTEMS WITH INTEGRATED SUPPORT

BVS/CMS

Infosoft's tape management system BVS for VM is equipped with an integrated interface to HACC/VM. This is defined when BVS is configured.

During cartridge processing in a CMS machine, a mount request is transmitted to the HACC/VM system by the BVSOPEN routine using a CP SMSG command.

An important aspect for the configuration of the HACC/VM system is that all CMS users which intend to process cartridges of the robot system via BVS must be registered as HACC/VM subsystems in the respective HACC/VM parameter file (S04$SUBS).

Management of the drives of the robot system is entirely under the control of BVS. BVS organizes the necessary dismount/keep of the drives when tape processing is completed by the CMS user.

INDIRECT INTEGRATION VIA HACC/VM ROUTER

DYNAM/T-CMS AND DYNAM/B

A HACC router machine must be set up for the integration of DYNAM/T into HACC/VM (see *HACC/VM Installation and Customization*). This router machine controls the necessary DYNAM/T scenarios that enable automated tape management with the AML system. The following strategies are applied:

- DYNAM/T-CMS manages only drives which are free from the perspective of DYNAM/T (CP FREE). Therefore, cartridges that are mounted for DYNAM/T-CMS by HACC/VM have to be set into the status CP FREE. This is done by the router machine (using the HACC/VM command RESERVE - see *HACC/VM Operator Guide*) when a mount request is generated by DYNAM/T-CMS.

- On completion of tape processing the respective drives are returned to the automatic drive management of the HACC/VM system by the router machine (using the HACC/VM command RELEASE - see *HACC/VM Operator Guide*).

VM:TAPE AND VM:BACKUP

A HACC router machine must be set up for integrating VM:Tape into HACC/VM (see *HACC/VM Installation and Customization*). This router machine controls the necessary VM:Tape scenarios that enable automated (and manual) tape processing by the AML system. The following strategies are applied:

- The VM:Tape SETUP facility and the AUTOPICK feature must be activated (VM:Tape configuration).

  AUTOPICK VM:Tape automatically demands a specific scratch volume for scratch requests. When such a specific volume is denied by the VM:Tape command REJECT, VM:Tape automatically selects another scratch volume.

  SETUP Before VM:Tape assigns a drive address to a certain requested volume (scratch or specific), the respective mount request is placed in a waiting queue when using the SETUP facility. Only after releasing this mount request by the VM:Tape command RELEASE or by assignment of a drive address using the VM:Tape command CHANGE is processing by VM:Tape resumed.

  In this way, HACC/VM can determine, prior to the selection of an available drive, whether the requested cartridge is present in the AML system. When the requested volser is found in the robot system, it is mounted on an available drive by HACC/VM. Subsequently VM:Tape is informed (via the VM:Tape commands CHANGE) which drive address should be used.

- In addition to the characteristics mentioned above, the following definitions must be present in the different configuration files of VM:Tape or VM:Backup (here *hacrtr* is the Userid of the HACC/VM router machine):

  Adjustments in VMTAPE CONFIG:

  - SETUPOPR *hacrtr*
  - TAPEOPER *hacrtr*
  - DEVICE NAME HACC *rdev1 rdev2* ...
  - DEVICE NAME MAN *rdev1 rdev2* ...
  - AUTOPICK AUDIT 999 OUTCODE GONE ACCEPT ATTEMPTS 2 CANCEL
  - MESSAGE MSGNOH

Adjustments in VMBACKUP CONFIG:

- SYSOPER *hacrtr*
- TAPEOPER *hacrtr*
- RESERVE OFF
- TAPEDISP DETACH
- AUTHORIZ OPERATOR *hacrtr*

þ The following definitions must be present in the HACC/VM parameter files:

S04$SUBS

- *vmtape* VMT STD R N N *. 50 . Y Y 1 _TMSS R(*);V(R)

S02$OUSR

- ROUTER *hacrtr* INITSELF:DISABLE A10:A11:A12:A13,
  !! B10:B11:B15:C10:C20:C30:D10:E10:E11:F10:H10

The following figure depicts automatic tape processing using VM:Tape and the AML robot system:

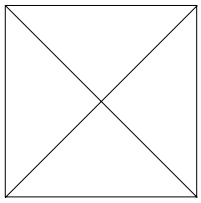


Figure 3: VM:Tape integration via HACC/VM router

The following figure depicts manual tape processing after integration of VM:Tape into the AML System:

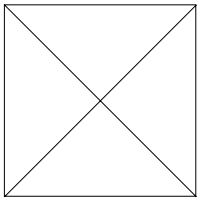


Figure 4: Manual operation of VM:Tape

Cartridges requested by VM:Tape can be processed automatically using the drives of the robot system, but also by using manually operated drives:

1. The **exclusive** manual mounting of a requested cartridge outside the robot system by an operator can be achieved when the option NOSETUP is used in the VM:Tape command MOUNT.
2. When the requested specific (not SCRATCH) cartridge is **not** in the robot system, it can be

   a) inserted through the insertion/ejection area of the robot system. HACC/VM recognizes this (via INSERT Exit) and automatically mounts the appropriate cartridge.

   b) mounted by an operator on a drive outside the robot system. If the respective manual drive for the HACC/VM router has not been started yet, the respective drive address has to be started by VMTSTART or VMTMOUNT.

DIRECT INTEGRATION

HACC/BR (BACKUP / RESTORE)

The data backup system HACC/BR is a “miniature“ tape management system for fully automatic backup and restoration of VM data sets specifically developed for use of the AML robot system.

Currently, DASD data backup is supported by the IBM program DDR.

Simultaneous backup of data sets on different drives is possible and significantly reduces the time necessary for backup.

INTRODUCTION INTO HACC/BR

Data backup and data restoration (Backup/Restore) within the environment of the operating system IBM/VM is largely determined by the VM-specific architecture of the hard disk. In particular, the decisive criteria in VM are:

- DASD architecture (CKD, FBA)
- Type of hardware equipment (3380, 3390,...)
- VM allocation (SPOL, PAGE, PERM, etc.)
- MDISK concept (physical start/end limitation)
- Different storage and data organization / MDISK
  - CMS - storage organization
  - SFS - storage organization
  - CP spool
  - other (e.g. VSAM, CATIA etc.)

According to the architecture-specific requirements and taking into account the safety demands of the respective VM operation (computing center), an adequate Backup/Restore procedure is needed for each particular storage organization.

This requirement is basically satisfied by the following Backup/Restore components which are part of the VM operating system delivered by IBM:

- DDR (DASD Dump/Restore) Program
  - Physical image of a hard disk

- SFS (Shared File System) Backup/Restore Utility
  - Image of VM-SMS-DB

- SPTAPE (VM spool) Backup/Restore Utility
  - Image of the VM-Spool-Dataset

- FILELEVEL (CMS) Backup/Restore Utilities
  - Image of CMS-MDISKs

Without using alternative Backup/Restore components (usually third party products) the VM user is committed to employing the VM components mentioned above. In this case B/R scenarios usually are processed as individual procedures.

From experience, such “isolated“ solutions often cover only the “very basic“ needs.

Major disadvantages of such commercial procducts are:

- no common platform or user interface
- no or insufficient support of standard labels
- no maintenance of backup data carriers or very difficult integration into computing center automation

An additional disadvantage is that some B/R products are very clumsy to use and/or are not suitable for automated applications.

HACC/BR offers all the backup procedures necessary under VM, including complete automation related to a robot system and appropriate schedule software.

THE CONCEPT OF HACC/BR

HACC/BR satisfies all the requirements for safe, transparent, and, if desired, fully automated execution of the Backup/Restore (B/R) scenarios typical for VM under central control and integration of an own catalogue maintenance.

HACC/BR is based on the VM standard procedures for backup and restore and uses exclusively official system interfaces and components available in the framework of the basic VM system. No additional program products are necessary.

Modification of the VM operating system software is not necessary for HACC/BR.

Installation and maintenance of HACC/BR is simple and is carried out by VM system administration.

HACC/BR has its own **catalogue maintenance**, where the data sets to be backed up and the respective attributes are defined in aggregate format:

- What to backup? (MDISK, FILEPOOL, SPOOL)
- How to backup? (DDR, CMSFILE, SFS, SPTAPE)
- Where to backup to? (3480, 3490,...)
- The backup time is organized by an appropriate scheduling procedure.

In the framework of VM directory maintenance and an appropriate **concept of directory definition** (page 45), it is already determined by HACC/BR, whether and how the "resource" *MDISK* is to be stored according to pre-defined aggregates.

For automatic operation a scheduling machine is recommended, which sends certain time-dependent backup requirements as HACC/BR commands per CP SMSG to the HACC/BR system. The syntax of the HACC/BR commands is described in the *HACC/VM Operator Guide*.

The HACC/BR system consists of the following components that must be installed as virtual VM/CMS machines:

BRM: (**B**ackup/**R**estore **M**anager) - server machine of the HACC/BR system, which controls all Backup/Restore procedures in HACC/BR. A detailed description of the BRM server machine can be found in *BRM - Backup/Restore Manager* on page 42.

MAM: (**M**edia **A**rchive **M**anager) - information management of the data sets processed in HACC/BR. The MAM machine is described in chapter *MAM - Media Archive Manager* on page 43.

BRP1...BRPn : (**B**ackup/**R**estore **P**rocessor) - an independent virtual machine defined for a B/R process and started by HACC/BR. Depending on the necessary resources (drives etc.), parallel execution of several B/R procedures is possible by the simultaneous use of several BRP machines. The BRP machines are described in chapter *BRP - Backup Restore Processor* on page 44.

HACC/BR SYSTEM OVERVIEW

HACC/BR is designed for fully automated applications. When certain interface requirements are fulfilled, all requests to the HACC/BR system can be executed automatically.

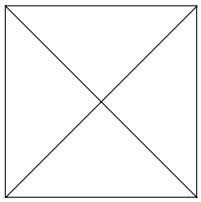


Figure 24:

The following interfaces integrate the HACC/BR system into the configuration of the VM computer:

- **SCI** (**S**chedule **C**all **I**nput) — Each backup procedure is initiated by a *Schedule Call*. In automatic operation this function is realized by the scheduling machine of the respective VM installation.
- **ROM** (**Resource O**peration **M**anagement) — The level, at which assignment of cartridge drives and the mounting of requested scratch or specific cartridges takes place. In the non-automatic mode this is usually a human operator, however, in automatic mode this is performed by a robot system.

· **BRM** (**HACC/BR M**anagement)	This is a user interface (commands and menus) for control and configuration of HACC/BR.

To implement an entirely automated backup scheme, the *Schedule Call* should be initiated by the scheduling machine used in the respective computing center operation. On the other hand, a cartridge robot system is needed to automate the mounting of cartridges.

BRM - BACKUP/RESTORE MANAGER

This component is the server machine of the HACC/BR system. All backup and restore requests are transmitted as commands (via CP SMSG) to this virtual machine.

HACC/BR is a “virtual“ multi-tasking system, were several Backup/Restore procedures are executed in parallel operation. Each request (command) sent to the HACC/BR system receives an unique identification label (command objectId). Under this *ObjectId*, the B/R request initiated by the command is managed in the **command queue** of the BRM machine. Depending on the particular backup or restore procedure, one or several tasks are generated which are sometimes processed in paralell independently from each other. The tasks are managed in the **task queue**.

When a B/R job (command) is completed, the respective command object is automatically removed from the command queue.

Special HACC/BR commands exist for status indication and control of the HACC/BR tasks or command objects (command and task queue), (see HACC/VM Operator Guide).

In principle, one cartridge (tape) drive as well as one BRP machine are assigned to each HACC/BR task.

In respect to the BRP machines the HACC/BR server machine BRM is working in the SCIF mode (secondary console interface). This means that all activities of the BRP machine are controlled by BRM. The more BRP machines are defined in the HACC/BR complex, the more backup (or restore) procedures can be processed in parallel. Of course, another requirement is a sufficient number of cartridge drives.

Under certain conditions (e.g. no drive and/or no BPR machine available) requests can be put on hold. As soon as the necessary resources are available they are automatically reactivated.

MAM - MEDIA ARCHIVE MANAGER

The media archive manager (MAM) component of the HACC/BR system is the central catalogue manager of all backup data carriers and aggregate definitions maintained under the control of HACC/BR.

All information regarding backup procedures executed under the control of the BRM machine are catalogued and managed by the MAM component.

The MAM component manages the following information and its interrelationships :

- Aggregate (see *Aggregate* on page 46)
- HACC/BR cartridges and tapes
- backup set information

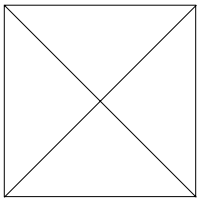


Figure 25: MAM catalogue management

BRP - BACKUP RESTORE PROCESSOR

The BRP component of the HACC/BR system may exist several times. In order to use all cartridge stations of a connected robot system with HACC/BR/VM, the number of defined BRP machines and the number of cartridge stations should be identical. During the configuration phase after installing HACC/BR or later, one or several virtual machines are defined as BRP components.

As a rule, one BRP machine is assigned to each active Backup/Restore procedure (HACC/BR Task), which executes the actual I/O processing under control of the BRM component. BRM controls the B/R procedures in the respective BRP machine via the SCIF (secondary console interface) protocol. The BRP machines are started and stopped automatically by the BRM component upon demand.

The number of defined BRP machines also limits the number of B/R procedures that can be executed simultaneously.

DIRECTORY DEFINITION CONCEPT

The minidisks of a VM system which need to be backed up by HACC/BR can be partitioned into logical units using the HACC/BR directory definition concept. For instance, all minidisks that need to be backed up on one particular day can be one logical unit, whereas all minidisks that need to be backed up on a weekly basis may form another unit .

HACC/BR allows the assignment of different attributes to these logical units and to back them up automatically.

To backup particular minidisks of individual VM users physically (DDR) or logically (VMFPLC2), synonyms (abbreviations) are assigned in the VM directory to the minidisks belonging to one backup. Then, these synonyms are assigned to defined aggregates in the HACC/BR, which in turn can be specified with attributes:

e.g.: **DD** - **D**aily **D**DR
WD - **W**eekly **D**DR
MD - **M**onthly **D**DR

A HACC/BR synonym is assigned to a minidisk by appending the respective minidisk definition with the name of the synonym:

MDISK *vaddr volser start end mode rpw wpw mpw <synonym>*

Alternatively, the HACC/BR program ABRXDIR can be started after each directory update. All information necessary for HACC/BR operation is extracted from the VM source directory and transmitted in a spool file to the BRM machine of the HACC/BR system. The information file generated during this process contains user and minidisk information only, no passwords.

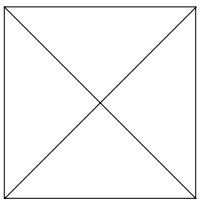


Figure 26: Directory definition concept

AGGREGATES

A basic term of the HACC/BR system is the aggregate. Under this term a variety of functions and attributes of a HACC/BR backup procedure can be summarized and requested. In other words, an aggregate defines, how to backup which data and which accompanying procedures must be executed.

Examples:

- all minidisks labelled with the synonym DD in the VM directory have to be backed up with DDR daily. For this purpose the synonym DD is assigned to the HACC/BR aggregate *DDR.DAILY* and is defined to keep 7 versions, i.e. when the 8th version is generated all cartridges (tapes) of the oldest version should be automatically released and should become available again as scratch cartridges.
- when defining a DDR backup of a Shared File System (SFS) data set in the HACC/BR aggregate *SFSDDR.SHARED.DAILY* (not to be mistaken for a SFS backup with an available SFS procedure), it is specified that the respective SFS server is stopped to warrant data integrity, that all minidisks of the file pool SHARED are backed up and that subsequent to the backup the SFS server is started again.

Furthermore, aggregates can be defined such that certain data sets which need to be backed up will be updated on cartridges for optimum utilization of tape material.

Example:

- minidisk 191 of the PROP machine is backed up daily and has to be updated on the same cartridge as the backup of the previous day. For this backup the aggregate *DDR.PROP.191* is defined and supplemented with the attribute *CONTINOUS*. In addition, the UserId of the PROP machine and the minidisk address 191 are assigned to this aggregate.

All data sets backed up by HACC/BR are catalogued by the MAM component, so that during a later restore all valid backup versions are accessible. Therefore, each minidisk backed up by HACC/BR (in case of DDR backups) or each backed up CMS file (in case of CMSFILE or SFS backups) can be restored.

The following aggregates are pre-defined during a successful installation and can be used as is, supplemented, changed or deleted:

Aggregate Name	Map Name	# of Versions	Comment
DDR.DAILY	DD MAP	7	daily DDR MDISK backup
DDR.WEEKLY	DW MAP	4	weekly DDR MDISK backup
DDR.MONTHLY	DM MAP	3	monthly DDR MDISK backup
DDR.QUARTERLY	DQ MAP	4	quarterly DDR MDISK backup
DDR.YEARLY	DY MAP	3	yearly DDR MDISK backup
CMS.DAILY	CD MAP	7	daily CMS File backup
CMS.WEEKLY	CW MAP	4	weekly CMS File backup
SFS.DAILY	SD MAP	7	daily SFS backup
SFS.WEEKLY	SW MAP	4	weekly SFS backup
SPTAPE.DAILY	SPD MAP	7	daily Spool backup
SPTAPE.WEEKLY	SPW MAP	4	weekly Spool backup

DDR BACKUP

An important component of HACC/BR is the physical (in contrast to logical) data backup using the DDR program which is common in the VM operating system environment. Here, certain DASD areas or entire hard disks are backed up on tape without consideration of their contents.

The DDR backup under control of the HACC/BR is fully automated. When used together with the HACC/VM robot host software[1] no further human intervention is needed. Selection of drives and mounting of the respective cartridges is performed by HACC/VM. The later control and catalogueing of the DDR backup run is performed by HACC/BR.

Besides the regular DDR backup, using disk addresses and cylinder areas (or block areas in case of FBA diks), HACC/BR offers the following options of DDR disk backup:

1. USER MDISK — individual VM user minidisks can be backed up
2. AGGREGATE — backup of pre-defined aggregates (logically connected data sets)

In case of DDR backup, HACC/BR also uses IBM standard labels to prevent access to foreign cartridges which have not yet been released.

Furthermore, at the beginning of each DDR backup a layout file is written on the cartridge as a second tape file next to the label to enable the identification of the backed up data set by displaying the file contents in case of an offline restore (without HACC/BR).

```
INPUT devno type (SKIP 1
DDR TYPE GRAPHIC
```

However, during an offline restore it should not be forgotten to skip the first two tape files before starting the RESTORE.

```
INPUT devno type (SKIP 2
```

[1] If HACC/BR is not connected to a robot system, the operator is asked, to mount the respective cartidges (tapes) manually, based on the scratch-list provided by HACC/BR.

CMS FILE BACKUP

A CMS file backup is defined as a logical backup on the CMS data level. CMS files written on mini disks and those managed by the Shared File System (SFS) are distinguished. Usually, the backup is carried out by the CMS Utility VMFPLC2.

HACC/BR supports the following backup procedures for the logical backup of CMS data sets:

1. FULL — Backup of a complete data set on minidisk or SFS directory
2. INCREMENTAL — Subsequent to a full backup of the data set of a CMS minidisk or a SFS directory, *incremental backups* can be made. In this case only those data are backed up which have changed since the last backup or those that have been added. This allows to maintain a relatively large number of backup versions with a small memory space requirement.
3. AGGREGAT — As in case of the DDR procedure it is possible to perform backups of aggregates.

A logical backup also uses HACC/BR IBM standard labels.

The logical backup of CMS files will be supported in the next release .

SFS (SHARED FILE SYSTEM) BACKUP

Shared File System (SFS) backup will be supported in the next release.

SPTAPE BACKUP

SPTAPE backup will be supported in the next release.

PERFORMANCE CHARACTERISTICS

- Own catalogue management (MAM). The **M**edia **A**rchive **M**anager stores all the necessary and essential information related to Backup/Restore procedures.

  This data is stored in a transparent form in the ’Media Archive Catalogue’. The catalogue can be manipulated interactively with an authorized CMS machine.

- Backup scenarios for minidisks are described in association with the VM directory.

  Together with a directory update an updated list of the minidisk data sets assigned to the respective backup aggregates is generated automatically.

- Aggregate backup

  Sum of the logical data set

- Multi-file volumes, Multi-volume files are supported

- Standard label (OS) processing

- Expiration date verification

- Interface: HACC/VM (AML)

REQUIREMENTS

VM/ESA Release 1 and higher.

HACC/BR - INTERNAL COMMUNICATION

BRM DEMANDS ON MAM

OPEN (AGGREGATE) REQUEST

OPEN { **AGG** *Aggregate* / **SYN** *Synonym* } [**(TID** *tid*]

The Open request is sent to MAM either together with the full aggregate name (AGG *Aggregate*) or with the aggregate synonym (SYN *Synonym*).

If an Open is requested for an aggregate which is not yet defined in MAM, this aggregate is created automatically using standard definitions.

MAM response: As a response BRM expects the message MAMttt100 and possibly one or more MAMttt105I messages for currently existing streams of the aggregate.

tid **MAM**ttt**100**{ **I** / **W** } *Aggregate,Synonym,Owner,CDate,EDate,Streams*

Tid	BRM Task-Id
Aggregate	the full aggregate name of the aggregate
Synonym	the synonymous name of the aggregate
Owner	the Owner ID of the aggregate
CDate	Creation date of the aggregate
EDate	Expiration date of the aggregate
Streams	Number of existing data streams and therefore the amount of information that still needs to be transmitted (MAMxxx105I).

If the value *Streams* returned by MAM, is greater than 0, further messages of the type MAMttt105 are expected.

tid **MAM**ttt**105I** *StreamId Volser Fileno*

Tid	BRM Task-Id
StreamId	The logical designation of a data set that exists on a cartidge/tape.
Volser	The volser where the respective data set is to be updated (StreamId from aggregate command file).
Fileno	The number of the last tape file after which the update should be performed.

OPENVOL (OPEN VOLUME) REQUEST

OPENVOL *Vol* **AGG** *Aggr* **TYPE** { **R**ead | **W**rite } [**(TID** *Tid* **LABREC** *Lblrec*[)]]

When a tape requested for a backup or restore is mounted, it is opened by the OPENVOL request. For this purpose the tape label is read and verified.

Tid	BRM Task-Id
Vol	Volser of the cartridge to be examined
Aggr	Name of the aggregate for which this volume is to be opened
Lblrec	The label information to be verified
1-6	tape label
7-16	owner
17-33	dataset name (aggregate name)
34-39	DSN volser (first tape label in case of multi-volume DSN)
40-43	volume sequence number (in case of multi-volume DSN)
44-49	creation date
50-55	expiration date

MAM response: The message MAMttt101 is expected as a response to an OPENVOL request. The message MAMttt101W indicates that an error occured during opening.

tid **MAM**ttt**101** { I | **W** } *Text*

CLOSEVOL (OPEN VOLUME) REQUEST

CLOSEVOL *Vol* **AGG** *Aggr* [**(TID** *Tid*[)]]

When the processing of a cartridge by BRM has been completed successfully, a CLOSEVOL request is sent to MAM.

Tid	BRM Task-Id
Vol	Volser of the cartridge to be examined
Aggr	Name of the aggregate for which this volume is to be opened

MAM response: Confirmation for CLOSEVOL request

tid **MAM**ttt**108**{**I** | **W**} *Text*

UPDAGG (UPDATE AGGREGATE) REQUEST

UPDAGG *Aggr* **VOL** *Vo*l **STREAM** *Id* **FILENO** *n* **(TID** *Tid* **INFO** *Info*[)]

Via the UPDAGG request, MAM receives aggregate information for catalogueing.

Tid	BRM Task-Id
Aggr	the aggregate name to be used for catalogueing the information.
Id:	the StreamId for information catalogueing
Info	*Infotyp Inforecord*
	Info record for infotype DDR:
	*U userid vdev User Minidisk Information
	or
	*A alloctype DASD Allocation Information
	Rdev real hard disk address
	Typ hard disk decive type (3380, 3390 etc.)
	Dvol DASD Volser
	Scyl Start Cylinder
	Ecyl End Cylinder

MAM response: Confirmation for UPDAGG request

tid **MAM**ttt**102**ì **I** ü
í ý *Tex*t
î **W** þ

CLOSE (AGGREGATE) REQUEST

CLOSE *Aggregat* [**(TID** *tid*]

MAM response: Confirmation for CLOSE request

tid **MAM**ttt**103**ì I ü
í ý *Text*
î **W** þ

CANCEL (AGGREGATE) REQUEST

CANCEL *Aggregat* [**(TID** *tid*]

MAM response: Confirmation for CANCEL request

tid **MAM**ttt**104**ì I ü
í ý *Text*
î **W** þ

UPDVOL (UPDATE VOLUME) REQUEST

UPDVOL *Vol* **TYPE** { **UPDATE** **AGG** *Aggr* / **INIT** } [**(TID** *tid* **LABREC** *Lblrec.*]

When initializing a scratch cartridge or writing on it for the first time, the MAM is supplied with the respective label information using the UPDVOL request.

Vol	Volser of the cartridge for which label information is to be updated in the catalogue
Aggr	Name of the aggregate used for this volume
Tid:	BRM Task-Id
Lblrec.	As in CHKVOL

MAM response: Confirmation for UPDVOL request

tid **MAM**ttt**106** { I / **W** } *Text*

LCKVOL (LOCK VOLUME) REQUEST

LCKVOL *TapeVol*

In the event a writing error occurs, the cartridge is disabled for further use by the LCKVOL request. After examination, this cartridge may be re-activated by a new initialization.

MAM response: Confirmation for LCKVOL request

tid **MAM**ttt**107**ì **I** ü
í ý *Text*
î **W** þ

MAM DEMANDS ON BRM

INIT

Cartridges intended for use by the HACC/BR system are initialized using the INIT command.

INIT *Volser*

Volser the volser under which the cartridge is maintained in HACC/BR

ADSTAR DISTRIBUTED STORAGE MANAGER (ADSM)

Integration of IBM's backup and archive system ADSM is performed by the sample REXX procedure DSMMOUNT EXEC which is delivered with HACC/VM. This procedure has to be adjusted by the customer (Userid of the HACC/VM server machine) and must be made available for the ADSM mount machines DSMEXITn.

The ADSM server machine (DSMSERV) must be defined as an HACC/VM subsystem (S04$SUBS PARM), the ADSM mount machine must be defined as HACC/VM operator (S02$OUSR PARM).

GLOSSARY

This glossary defines the most important HACC/VM and HACC/VSE abbreviations and terms.

$JOBEXnn. This SVA routine performs recovery tasks as a HACC/VSE component after job aborts and at the end of jobs and steps when VSE systems are connected to HACC/VM.

$ROBEXIT. This SVA routine performs the communication with the HACC/VM system as a HACC/VSE component when VSE systems are connected to HACC/VM.

AML. Automated Media Library. Identifies cartridge robot systems made by ADIC/GRAU Storage Systems.

Adapter. The HACC/VM adapter machines serve to communicate with the AML systems.

Aggregate. VM data stock defined in the MAM archive that can be automatically backed up with HACC/BR.

Alert. A so-called Alert UserId is defined within the HACC/VM system parameters and this UserId is alerted by messages or spool files when problems occur during automatic operation. An alert log (*svrid* ALERTLOG) is also created for problem analysis.

AMU. AML Management Unit. the control computer of an AML system. Also used for communication between HACC/VM and the AML System.

Archive. All coordinates and cartridges (volser) are stored in the AMU database (SQL). An archive can also be kept in the HACC/VM system. This is especially necessary when several HACC systems (HACC/VM and HACC/MVS) access an AML system without organizational separation.

Batch Process. HACC/VM creates a batch process for each Batch_Command_File received with an own task number (TaskId). Special HACC/VM commands can be used to control batch processes.

Batch_Command_Facility. A HACC/VM facility to execute certain organizational processes (i.e. ejecting a large number of cartridges) by sending several HACC/VM commands simultaneously in a CMS file.

Batch_Command_File. A file (file name BTCHCMD) containing AML commands (*BAT statements) to be processed by HACC/VM. A Batch_Command_File is sent to the virtual reader of the HACC/VM server machine for processing.

The sender normally receives an acknowledgement as a reply list after the Batch_Command_File has been processed.

BR (**B**ackup/**R**estore). Optional HACC/VM components for automatic backup in VM. Can be of use when VM does not have a tape management system.

BRM (**B**ackup/**R**estore **M**anager). Subcomponent (virtual machine) of BR. Controls all backup and restore tasks of the BR.

BRP (**B**ackup/**R**estore **P**rocessor). Subcomponent (virtual machine) of BR. Several BRP machines can be used to process several BR backup and restore tasks in parallel.

Cleaning. The write/read heads of the cartridge drives must be cleaned from time to time by inserting a cleaning cartridge. This is a preventative measure in HACC/VM, this means that HACC/VM automatically mounts a cleaning cartridge before the control unit of the cartridge demands cleaning.

Client. HACC/VM regards all virtual machines that communicate with HACC/VM server as clients. Client types are grouped as follows:

- SUBSYSTEM
- ADAPTER
- OPERATOR
  - ROUTER
  - MONITOR

Dynamic area. An area defined as dynamic has no specific assignment between the volser of a cartridge and a coordinate as in a hierarchic system. This means that the respective slot released by the removal of a cartridge within the dynamic area can be used later for the insertion of a different cartridge.

DYNEXIT. The following exits are implemented in this routine when DYNAM/T is used as tape management system under VSE:

- Message Exit MSGHOOK
- Open Exit OPNHOOK
- Close Exit CLSHOOK

EJECT. EJECT is the process of cartridge removal by the insertion/ejection unit of the AML system.

EPIC. Tape management system from the Legent company.

FAQS/ASO. Software package from the Legent company for automatic control of a VSE system.

FMSD. Foreign Mount Source Device. Refer to foreign mount.

Foreign mount. Cartridges that are to be loaded temporarily on a drive within the AML system can be loaded using the so-called foreign mount area of the input/output device. The AML system does not check or consider a barcode which may be on the cartridge.

HACC (Host Control Component). Control software that controls the connection between host applications and the AML robot system.

HACC/VSE. Interface software that is implemented when connecting the VSE tape management systems DYNAM/T and EPIC/VSE.

HACCPARM. A parameter file to be assembled for the connection of DYNAM/T-CMS to HACC/VM. The HACCPARM EXEC procedure must also be adapted when the HACC/VM router machine is used.

HACCPVSE. The HACCPVSE parameter file is assembled and linked in the VSE system when DYNAM/T-VSE is connected to HACC/VM. The corresponding phase (HACCPVSE.PHASE) must be loaded in the SVA when HACC/VSE is enabled.

HACCVSNAP. This internal routine automatically generates a complete list of all significant variables when a HACC/VM error occurs.

Hexa tower. A revolving storage rack within an AML system comprising 6 segments. The maximum load capacity is currently 4320 cartridges.

IMOD (intelligent module). Identifies REXX procedures that can be executed under control of the FAQS/ASO software of the Legent company in a VSE system.

INSERT. INSERT is the process of cartridge insertion by the insertion/ejection unit of the AML system.

KEEP. The request to HACC/VM to dismount a cartridge from a drive supported by the AML system.

Coordinate. A unique coordinate is assigned to each slot serviceable by the AML system as well as every drive.

Logging. All messages received by HACC/VM are recorded in the SERVRLOG LOG1 log file and all messages sent by HACC/VM are recorded in the SERVRLOG LOG2 log file. (Enabled during HACC/VM parameter settings).

MAM (**M**edia **A**rchive **M**anager). Subcomponent (virtual machine) of BR to manage backup data files created with BR.

MLOG. All tasks sent to the HACC/VM system as SMSG messages are managed in the so-called message log (message queue) and are moved to the TLOG as soon as all resources required for execution are available (drive, volser).

Monitor. The HACC/VM monitor function serves to control and monitor the HACC/VM system. A HACC/VM monitor machine is a special HACC/VM Operator.

MOUNT. The request to HACC/VM to load a cartridge on a drive supported by the AML system.

Operator. A HACC/VM operator is a virtual machine that can generate HACC/VM commands. A special operator is a HACC/VM monitor machine.

Problem box. When any sort of mechanical problems occur during cartridge handling in the AML system, the cartridge is ejected to the problem box.

Quadro tower. A revolving storage rack within an AML system comprised of 32 segments with 4 inner towers (each with 6 segments) and 8 outer segments. The maximum load capacity is currently 5760 cartridges.

Robot. One or two robots within an AML system that service the cartridges and drives within the AML system.

Router. A HACC/VM router machine can generate HACC/VM commands controlled by console messages (also SMSG) from another virtual machine or by CP messages. The HACC/VM router machine is used, for example, when DYNAM/B is connected. The HACC/VM router machine is a special HACC/VM operator type.

Scratch_Facility. A HACC/VM facility that supports scratch substitution. Internal scratch lists are created from scratch lists read in from the respective tape management system via the virtual reader of the HACC/VM server machine.

Scratch substitution. HACC/VM supports the scratch mount request. For this purpose, HACC/VM manages internal scratch lists generated from the respective lists of the tape management system.

Server. The HACC/VM server machine manages all tasks passed to the AML system.

Stealing. Tape management systems with the AVR function (Automatic Volume Recognition) do not create a mount message when a requested cartridge is already loaded on an available drive. This means that a drive is assigned to an application with AVR for which a parallel application has already created a mount request (i.e. by scratch processing). This leads to two applications requiring a cartridge, but only one mount request. HACC/VSE recognizes this situation and automatically creates a second mount request in this case.

TLOG. The so-called task log contains all active HACC/VM tasks.

TMS exit. An interface application that passes certain information to HACC/VSE via the tape management system (i.e. messages). This exit is part of the HACC/VSE in certain circumstances.

TMS. Tape Management System (tape management system such as Dynam/T or BVS).

Glossary